

Adina Butnărașu

Programarea liniară. Algoritmul Simplex

Editura PIM
IAȘI, 2009



Adina Butnărașu

Programarea liniară. Algoritmul Simplex

Dacă sunt greșeli,
peste m. de "perzi"
discret...

Adina Butnărașu

Adina Butnărașu este absolventă a Facultății de Matematică a Universității „Al. I. Cuza” din Iași, promoția 2000. După nouă ani la catedră și o licență în informatică (2009), apare ideea publicării acestei cărți, ca rod al studiului în cele două domenii care i-au adus atâtea satisfacții profesionale: matematica și informatica.

EDITURA pim

Soseaua Stefan cel Mare nr. 11 Iasi -700498

Tel. / fax: **0332-440728**

e-mail: editurapim@pimcopy.ro

www.pimcopy.ro

EDITURĂ ACREDITATĂ CNCIS BUCUREȘTI

66/01.05.2006

ISBN: 978-606-520-566-6

© Tehnoredactare, profesor Adina Butnărașu

Toate drepturile asupra acestui material didactic sunt rezervate autorului
Nici o parte a acestuia nu poate fi reprodusă fără acordul scris al autorului



Adina Butnărașu

Programarea liniară. Algoritmul Simplex

IAȘI, 2009
Editura pim

CUPRINS

| | |
|--|-----|
| Cuvânt înainte | 7 |
| Introducere | 11 |
| Capitolul I Programarea liniară | 15 |
| Capitolul II Modele tehnico-economice de optimizare ce conduc la o problemă de programare liniară | 45 |
| Capitolul III Algoritmul Simplex | 59 |
| Capitolul IV Programe pentru rezolvarea problemelor de programare liniară | 83 |
| Bibliografie | 105 |

CUVÂNT ÎNAINTE

Activitatea economico-socială modernă se caracterizează prin dinamism, concurență și eficiență. Pentru a rezista pe o piață în continuă schimbare, care solicită produse și servicii cât mai performante, mai diversificate și mai complexe la prețuri cât mai reduse sunt necesare eforturi sporite în direcția optimizării resurselor de care dispune o anumită organizație, în condițiile asigurării îndeplinirii scopului acesteia la o calitate impusă.

Asigurarea acestor cerințe, care pot părea contradictorii, necesită un *management cu pregătire superioară*, bun cunoscător a cerințelor grupurilor țintă (un marketing agresiv în sensul bun al cuvântului), care să creeze modele cât mai apropiate (inclusiv în timp) de fenomenele economico-sociale analizate și cărora să le aplice metode matematice corespunzătoare, bazate pe utilizarea avansată a tehnicii de calcul. Nu în ultimul rând, conducerea organizației trebuie să fie capabilă să interpreteze rezultatele, să formuleze decizii corespunzătoare (care să prezinte un minim de risc) și să urmărească aplicarea lor.

Formarea acestor manageri trebuie începută din școală, pe măsură ce dobândesc cunoștințe adecvate. Trebuie învățați să creeze modele pornind de la o problemă, cărora să le aplice o metodă în vederea obținerii unei soluții (optim).

Programarea liniară face parte dintre problemele cercetării operaționale, direcție a matematicii a cărei denumire a fost formulată la sfârșitul ultimului război mondial. Algoritmul care permite rezolvarea generală a problemei de programare liniară este algoritmul Simplex, conceput de matematicianul american G.B. Dantzig, în anul 1947.

Pentru aplicarea acestui algoritm sunt necesare cunoștințe de calcul matricial, pe care elevii le capătă în liceu.

Cartea de față prezintă formularea problemei de programare liniară, tipuri de probleme care se pot modela și rezolva cu ajutorul acestei metode, algoritmul Simplex și utilizarea tehnicii de calcul pentru aplicarea algoritmului Simplex la problemele de programare liniară.

Autorul lucrării, d-na profesor Adina Butnărașu, are meritul de a fi descris succint și clar, pe înțelesul elevilor, teoria legată de formularea și rezolvarea problemei de programare liniară. Un accent deosebit este pus pe descrierea unor aplicații din viața economico-socială a căror optim trebuie găsit. În acest fel, elevii sunt puși să creeze modele matematice adecvate cărora să le aplice metoda de programare liniară și ulterior să interpreteze rezultatul obținut.

Această prezentare a problemei de programare liniară permite înțelegerea mai bună atât a teoriei și a posibilităților ei de aplicare la rezolvarea unei multitudini de probleme de optimizare.

De asemenea, se remarcă prezentarea unui program descris în limbajul de programare Basic pentru rezolvarea problemei de programare cu ajutorul algoritmului Simplex.

Consider că prezenta carte constituie un sprijin real în activitatea didactică de predare la elevi a unor metode matematice avansate de optimizare.

Conf.dr.ing. Constantin Cărașu

INTRODUCERE

Nu mai constituie o noutate faptul că activitatea modernă în domeniul producției industriale și conducerii activităților economice a devenit atât de complexă, încât rezolvările tradiționale nu mai sunt suficiente. O mare parte dintre problemele care se pun în conducerea economică a întreprinderilor sunt probleme de optimizare.

Modelarea matematică a acestor probleme oferă posibilitatea determinării soluțiilor optime, contribuind pe această cale la ridicarea eficienței economice a întreprinderilor. În multe cazuri concrete problemele economice pot fi transformate în probleme matematice.

Lucrarea de față "Programarea liniară. Algoritmul Simplex Primal" își propune să aducă în prim plan problema programării liniare. Problema programării liniare constă în optimizarea, adică determinarea valorii maxime sau minime a unei expresii liniare denumită funcție obiectiv sau funcție scop, în prezența unor restricții exprimate prin ecuații sau inecuații, de asemenea liniare.

Lucrarea este împărțită în patru capitole.

Capitolul I – „Programarea liniară” – prezintă pe larg noțiuni fundamentale legate de programarea liniară, de scrierea unei probleme în formă standard sau canonică, de interpretarea geometrică a unei probleme de programare liniară. O interpretare geometrică a unei probleme de programare se poate obține simplu în cazul când problema

are numai două variabile și se prezintă sub forma canonică.

Orice problemă de programare liniară care conține numai două variabile se poate rezolva grafic. Deși lipsită de importanță practică, o astfel de rezolvare este foarte instructivă și permite utilizarea unui limbaj intuitiv comod, care se poate extinde la cazul general a n variabile.

Capitolul II – „Modele tehnico-economice de optimizare ce conduc la o problemă de programare liniară” pune în lumină importanța practică a programării liniare, atât pentru ingineri, matematicieni, economiști cât și tuturor celor care se preocupă cu rezolvarea problemelor de optimizare în diferite domenii ale științelor aplicative.

„Algoritmul Simplex”, metodă expusă în detaliu în capitolul III, constituie o metodă de explorare sistematică a programelor de bază mai precis de trecere de la un program de bază la altul, care dă funcției obiectiv o valoare mai mică (mare) pentru o problemă de minimizare (maximizare). Algoritmul furnizează de asemenea criterii pentru cazurile în care problema de programare liniară nu are programe de bază sau are optim infinit.

Tot în acest capitol sunt prezentate câteva exemple de rezolvare cu ajutorul algoritmului simplex a unor probleme de programare liniară de dimensiuni mici. Foarte important este că în practică numărul variabilelor sunt mult mai mari, ceea ce face dificilă rezolvarea “manuală” a problemelor de acest tip; în acest caz se face apel la calculatoarele electronice, care pot face calculele necesare mult mai

rapid și mai concret, putând rezolva ușor probleme de programare liniară în care apar sute de restricții sau de variabile.

Capitolul IV – „Programe pentru rezolvarea problemelor de programare liniară” reprezintă partea practică a lucrării. Programul în care s-au realizat rulările este scris în limbajul BASIC.

Cartea se adresează profesorilor care au în sfera preocupărilor predarea unor algoritmi de optimizare în cadrul unor activități didactice (de exemplu cercuri științifice ale elevilor).

Modelul matematic pentru programarea liniară e constituit dintr-un ansamblu de relații liniare dintre care una reprezintă obiectivul urmărit și se numește relația sau funcția obiectiv, celelalte cuprinzând restricțiile la obiectivul de împlinit.

Criteriul de optimizare ales impune în unele cazuri maximizarea funcției obiectiv, iar în altele, minimizarea acesteia. Variabilele care apar într-o problemă de programare liniară sunt de obicei supuse condiției de nenegativitate; există totuși cazuri în care unele dintre variabile nu sunt supuse la nici o condiție asupra semnului, iar în alte situații trebuie impuse condiții de nepozitivitate.

$$\max(\min)[Z(x)] = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ \phantom{a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n} \geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ \phantom{a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n} \geq b_2 \\ \dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\ \phantom{a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n} \geq b_m \\ x_j \geq 0 (j = 1, \dots, n) \end{array} \right.$$

în care:

x_1, \dots, x_n sunt variabilele reale;

c_1, \dots, c_n sunt coeficienții variabilelor din funcția obiectiv sau funcția de eficiență;

a_{11}, \dots, a_{mn} sunt coeficienții variabilelor de restricție;

b_1, \dots, b_m sunt termenii liberi ai restricțiilor.

Problema programării liniare se poate afla după formulare în așa zisă formulare primară, dar fiecare problemă își găsește și o formulare duală. În formularea primară se cere maximizarea funcției iar în formularea duală a acesteia se va cere minimizarea ei.

1. Forme de prezentare a modelului de programare liniară

FORMA CANONICĂ

O problemă de programare liniară este de formă canonică dacă toate restricțiile sunt inegalități de același sens și tuturor variabilelor li se impun condiții de nonnegativitate. Dacă sensul restricțiilor este " \geq " atunci funcția obiectiv trebuie minimizată iar dacă sensul restricțiilor este de tip " \leq " funcția obiectiv trebuie maximizată. În ambele cazuri restricțiile acestor probleme se numesc concordante.

Forma matematică a definiției de mai sus:

$$\min[Z(x)] = cx$$

$$\max[Z(x)] = cx$$

$$Ax \geq b$$

sau

$$Ax \leq b$$

$$x \geq 0$$

$$x \geq 0$$

în care:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \cdot & \cdot & a_{mn} \end{pmatrix} \text{ matricea coeficienților tehnologici;}$$

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ b_n \end{pmatrix} \text{ vectorul coloană, numit vectorul resurselor;}$$

$$c = \begin{pmatrix} c_1 \\ c_2 \\ \cdot \\ c_n \end{pmatrix} \text{ vectorul coloană, numit vectorul costurilor;}$$

$$c' = (c_1, c_2, \cdot, c_n) \text{ vectorul linie al costurilor;}$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{pmatrix} \text{ vectorul coloană al activităților.}$$

În aplicațiile practice apar frecvent situații în care modelul conține simultan restricții de toate tipurile.

Este clar că o problemă de programare liniară în forma generală poate fi adusă la forma standard, canonică sau mixtă folosind următoarele transformări echivalente ale problemelor de programare liniară:

- (a) sensul unei inegalități se schimbă prin înmulțirea cu -1 ;
- (b) transformarea inegalităților în ecuații: o inegalitate de forma $Ax \leq b$ poate fi scrisă ca o ecuație $Ax + y = b$, introducând o variabilă, numită variabilă ecart, variabilă abatere sau variabilă de compensare, $y \geq 0$, iar o inegalitate de forma $Ax \geq b$ se transformă în ecuația $Ax - y = b$ prin scăderea variabilei ecart $y \geq 0$;
- (c) transformarea ecuațiilor în inegalități: o ecuație de forma $Ax = b$ este echivalentă cu inegalitățile $Ax \geq b$ și $Ax \leq b$;
- (d) o variabilă supusă condiției de nepozitivitate ($x \leq 0$) se transformă într-o variabilă nenegativă prin substituția $x' = -x$;
- (e) o variabilă x căreia nu i impun condiții asupra semnului se poate înlocui cu două variabile nenegative x' și x'' prin substituția $x = x' - x''$;
- (f) deoarece totdeauna $\min f(x) = -\max(-f(x))$

O problemă de minimizare se transformă într-o problemă de maximizare și invers, schimbând semnele coeficienților din funcția liniară.

Rezultă că atât studiul teoretic cât și metodele de rezolvare pot fi abordate utilizând oricare dintre formele problemelor de programare liniară menționate.

APLICAȚII

1) Să se aducă la forma standard următoarea problemă:

$$\begin{cases} \max(3x_1 + 2x_2 - 2x_3) \\ 4x_1 + x_2 - 3x_3 = 1 \\ 2x_1 + 2x_2 - x_3 \geq 4 \\ 2x_1 - x_2 + x_3 \leq 2 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Prima restricție fiind egalitate rămâne neschimbată. Din prima parte a celei de-a doua restricții se scade variabila ecart $x_4^e \geq 0$ iar în restricția a treia se adună variabila $x_5^e \geq 0$, după care, se obține:

$$\begin{cases} \max(3x_1 + 2x_2 - 2x_3) \\ 4x_1 + x_2 - 3x_3 = 1 \\ 2x_1 + 2x_2 - x_3 - x_4^e = 4 \\ 2x_1 - x_2 + x_3 + x_5^e = 2 \\ x_1, x_2, x_3 \geq 0, x_4^e, x_5^e \geq 0 \end{cases}$$

2) Să se aducă la forma standard și la forma canonică următoarea problemă de programare liniară:

$$\begin{cases} \min(2x_1 - x_2 + 4x_3) \\ 2x_1 - x_2 = 10 \\ x_1 + 2x_2 \geq 1 \\ 2x_1 - x_2 - 2x_3 \leq -2 \\ x_1 \geq 0, x_2 \text{ oarecare}, x_3 \leq 0 \end{cases}$$

Forma standard:

$$\begin{cases} \min(2x_1 - x_4 + x_5 - 4x_6) \\ 2x_1 - x_4 + x_5 = 10 \\ x_1 + 2x_4 - 2x_5 - x_7 = 1 \\ 2x_1 - x_4 + x_5 + 3x_6 + x_8 = -2 \\ x_i \geq 0, 1 \leq i \leq 8. \end{cases}$$

S-a înlocuit variabila oarecare x_2 cu diferența a două variabile nenegative $x_2 = x_4 - x_5$, făcând substituția $x_3 = -x_6$ cu $x_6 \geq 0$ și introducând variabilele ecart x_7 și x_8 în cele două inegalități ale problemei.

Pentru a aduce problema la forma canonică, vom transforma prima ecuație în două inegalități de sens contrar; pentru ca toate inegalitățile problemei să fie concordante, vom înmulți cu -1 inegalitățile de tipul \leq . Făcând aceleași înlocuiri ale variabilelor x_2 și x_3 , obținem forma canonică:

$$\begin{cases} \min(2x_1 - x_4 + x_5 - 4x_6) \\ 2x_1 - x_4 + x_5 \geq 10 \\ x_1 + 2x_4 - 2x_5 \geq 1 \\ -2x_1 + x_4 - x_5 - 3x_6 \geq 2 \\ x_i \geq 0, 1 \leq i \leq 6. \end{cases}$$

3) Să se aducă la forma canonică problema:

$$\begin{cases} \max(2x_1 - 2x_2 + x_3) \\ 2x_1 + 3x_2 - 2x_3 \geq 1 \\ 2x_1 + 3x_2 - 2x_3 \geq 3 \\ x_1 + 2x_2 + 2x_3 = 4 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Prima restricție se înmulțește cu -1 iar cea de-a doua se înlocuiește cu inecuațiile

$$x_1 + 2x_2 + 2x_3 \leq 4$$

$$x_1 + 2x_2 + 2x_3 \geq 4$$

Se înmulțește cu -1 cea de-a doua inecuație. Se obține următoarea formă canonică:

$$\begin{cases} \max(2x_1 - 2x_2 + x_3) \\ -2x_1 - 3x_2 + 2x_3 \leq -3 \\ x_1 + 2x_2 + 2x_3 \leq 4 \\ -x_1 - 2x_2 - 2x_3 \leq -4 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

4) Să se aducă la forma standard și la forma canonică următoarea problemă de programare liniară:

$$\begin{cases} 2x_1 - x_2 = 10 \\ x_1 + 2x_2 \geq 1 \\ 2x_1 - x_2 - 3x_3 \leq -2 \\ x_1 \geq 0, x_2 \text{ oarecare}, x_3 \leq 0 \\ \min(2x_1 - x_2 + 4x_3) \end{cases}$$

Înlocuind variabila oarecare x_2 cu diferența a două variabile nenegative $x_2 = x_4 - x_5$, făcând substituția $x_3 = -x_6$ și introducând variabilele ecart x_7 și x_8 în cele două inecuații ale problemei obținem forma standard:

$$\begin{cases} 2x_1 - x_4 + x_5 = 10, \\ x_1 + 2x_4 - 2x_5 - x_7 = 1 \\ 2x_1 - x_4 + x_5 + 3x_6 + x_8 = -2 \\ x_1, x_4, x_5, x_6, x_7, x_8 \geq 0 \\ \min(2x_1 - x_4 + x_5 - 4x_6) \end{cases}$$

Pentru a aduce problema la forma canonică vom transforma prima ecuație în două inecuații de sens contrar; pentru ca toate inecuațiile problemei să fie concordante vom înmulți cu -1 toate inecuațiile de forma \leq deoarece problema este de minimizare. Făcând aceleași înlocuiri ale variabilelor x_2 și x_3 , obținem forma canonică:

$$\begin{cases} 2x_1 - x_4 + x_5 \geq 10, \\ -2x_1 + x_4 - x_5 \geq 1 \\ x_1 + 2x_4 - 2x_5 \geq 1 \\ -2x_1 + x_4 - x_5 - 3x_6 \geq 2 \\ x_1, x_4, x_5, x_6 \geq 0 \\ \min(2x_1 - x_4 + x_5 - 4x_6) \end{cases}$$

Concluzii:

Prin aducerea la forma standard se mărește numărul necunoscutelor problemei, iar prin obținerea formei canonice se mărește numărul restricțiilor problemei.

2. Restricții. Domeniile soluțiilor.

Funcția obiectiv, în cazurile obișnuite, are soluții cuprinse într-un domeniu definit de anumite limitări sau restricții. O definiție a restricției ar fi: expresia matematică care limitează domeniul posibil în care o soluție, pentru o problemă de optim, ar putea fi căutată.

Criterii principale de clasificare a restricțiilor:

- a. Linearitatea expresiilor: restricții liniare și neliniare
- b. Forma expresiilor: restricțiile pot fi de egalitate, de inegalitate și mixte.

Restricția liniară are forma:

$$g(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n \begin{matrix} \geq \\ \leq \end{matrix} b$$

Restricția va deveni mixtă când inegalitățile vor avea semnele: " \geq sau \leq ".

O interpretare geometrică a unei probleme de programare liniară se poate obține simplu în cazul când problema are numai două variabile. Orice problemă de programare liniară care conține numai două variabile se poate rezolva grafic; deși lipsită de importanță practică, o astfel de rezolvare este foarte instructivă și permite utilizarea unui limbaj intuitiv comod care se poate extinde fără mari dificultăți la cazul general.

Să considerăm problema de programare liniară în forma canonică:

$$\begin{cases} \max(x_1 + 2x_2) \\ x_1 \leq 2 \\ x_1 + x_2 \leq 3, \\ -x_1 + x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases}$$

Ecuațiile $x_1 = 2$, $x_1 + x_2 = 3$, $-x_1 + x_2 = 1$ reprezintă drepte în planul de coordonate Ox_1x_2 și impart planul în semiplane. Semiplanul $x_1 \leq 2$ determinat de dreapta (CD): $x_1 = 2$ este cel în care se află originea $O(0, 0)$. Analog, semiplanul $-x_1 + x_2 \leq 1$ determinat de dreapta (AB): $-x_1 + x_2 = 1$ este cel în care se află originea.

Semiplanul $x_1 + x_2 \leq 3$ determinat de dreapta (BC): $x_1 + x_2 = 3$ este de asemenea semiplanul care conține originea. Condițiile de nenegativitate $x_1 \geq 0$ și $x_2 \geq 0$ se reprezintă prin semiplane care conțin sensul pozitiv al axelor de coordonate $x_1 = 0$ și $x_2 = 0$.

Rezultă că punctele situate în interiorul sau pe laturile poligonului OABCD din figura 1 au drept coordonate soluțiile sistemului de inegalități:

$$\begin{cases} x_1 \leq 2 \\ x_1 + x_2 \leq 3 \\ -x_1 + x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases}$$

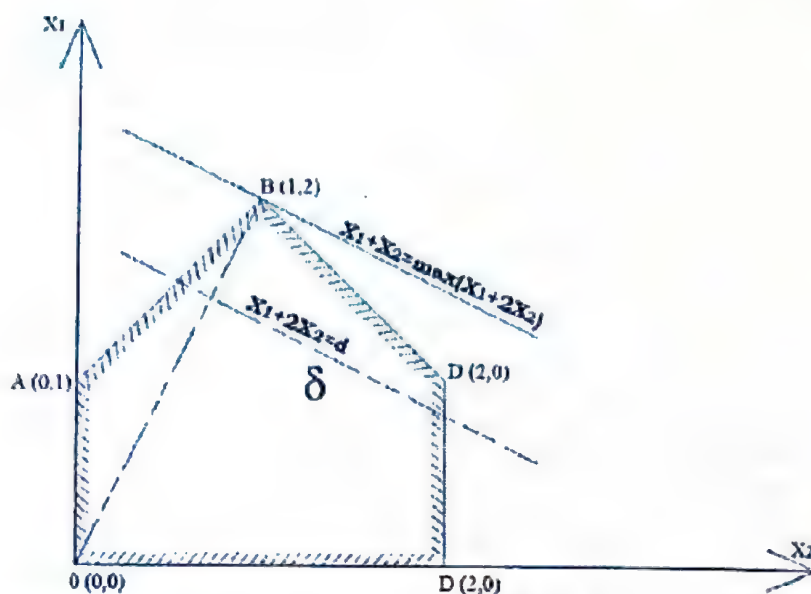


Figura 1. Se observă că soluția optimă a problemei de programare liniară este unul dintre vârfurile poligonului OABCD, respectiv punctul B de coordonate

$$x_1^* = 1, x_2^* = 2.$$

Dreapta de ecuație $x_1 + 2x_2 = d$ va fi numită curba de nivel a funcției obiectiv.

$$f(x_1, x_2) = x_1 + 2x_2.$$

Se știe că distanța δ de la $O(0, 0)$ până la dreapta $x_1 + 2x_2 = d$ este proporțională cu d .

Se vede imediat că cea mai mare valoare a lui d (adică a funcției obiectiv) se obține atunci, când distanța de la origine δ este maximă. Cum soluția optimă (x_1^*, x_2^*) verifică atât sistemul de inegalități cât și ecuația $x_1 + 2x_2 = d$ trebuie să aibă în comun cu poligonul OABCD cel

Programarea liniară. Algoritmul Simplex
 puțin un punct în așa fel încât δ să fie maxim. Aceste condiții sunt
 împlinite evident de coordonatele punctului B: $x_1^* = 1, x_2^* = 2$.

Se observă din acest exemplu că soluția optimă a acestei probleme de programare liniară este unul dintre vârfurile poligonului OABCD, care conține toate soluțiile sistemului de inegalități.

Dacă schimbăm funcția obiectiv $\max(x_1 + 2x_2)$ cu alta, este posibil ca soluția optimă a problemei să fie alt vârf al poligonului. În cazul în care curbele de nivel ale funcției obiectiv sunt drepte paralele cu una dintre laturile poligonului, soluțiile optime sunt în număr infinit, corespunzând punctelor de pe latura poligonului paralelă cu curbele de nivel ale funcției obiectiv.

Programarea liniară oferă domenii pentru soluțiile optime, domenii determinate de restricții.

În scopul exemplificării se iau următoarele restricții

$$x_1 + \frac{5}{4}x_2 \leq 5$$

$$x_1 + \frac{3}{5}x_2 \leq 3$$

Reprezentările grafice ale acestor inegalități într-un sistem de coordonate rectangulare x_1, x_2 sunt prezentate în figura 2:

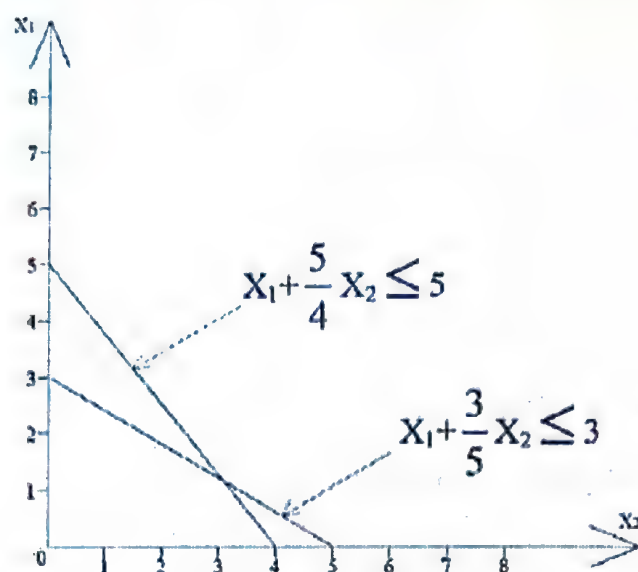


Figura 2.

Cele două drepte care reprezintă restricțiile determină un domeniu al soluțiilor posibile notat cu Δ_p și unul al celor imposibile notat cu Δ_n .

Dacă ar exista o restricție de egalitate de forma $2x_1 + 4x_2 = 16$, reprezentarea grafică a dreptei în sistemul de coordonate x_1, x_2 (figura 3) arată că sunt eliminate atât semiplanul de deasupra dreptei care poate fi descris prin inecuația $2x_1 + 4x_2 \leq 16$ cât și semiplanul de sub dreaptă, care poate fi descris prin inecuația $2x_1 + 4x_2 \geq 16$.

Posibilitatea înlocuirii unei restricții de egalitate prin două restricții mixte permite simplificări în prezentarea problemelor de optim.

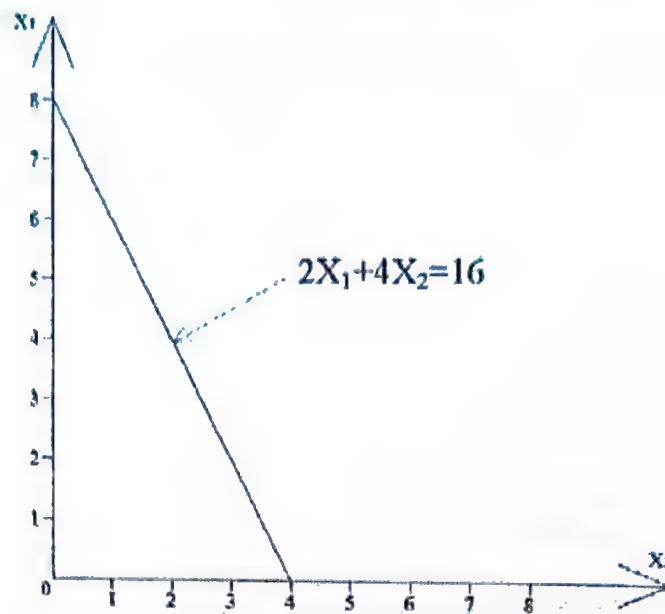


Figura 3.

În problemele de optimizare există, de obicei, un grup de restricții care odată cu creșterea numărului lor reduc tot mai mult domeniul posibil al soluțiilor.

Acum putem extinde problema interpretării geometrice la cazul general, mai exact, pentru o problemă cu un număr oarecare de restricții și numai două variabile. O astfel de problemă cere să se determine x_1 și x_2 așa încât

$$\left\{ \begin{array}{l} \min(\max)(c_1 x_1 + c_2 x_2) \\ a_{11} x_1 + a_{12} x_2 \leq b_1 \\ a_{21} x_1 + a_{22} x_2 \leq b_2 \\ \dots\dots\dots \\ a_{m1} x_1 + a_{m2} x_2 \leq b_m \end{array} \right.$$

Adina Butnărașu

unde \perp reprezintă una din posibilitățile $\geq, \leq, =$. De cele mai multe ori apar și condițiile de nenegativitate: $x_1, x_2 \geq 0$.

Ecuția de forma $a_{11}x_1 + a_{12}x_2 = b_1$ reprezintă ecuația unei drepte în planul x_1Ox_2 . Problema de programare liniară cere determinarea optimului funcției $f = c_1x_1 + c_2x_2$ pe mulțimea punctelor domeniului D, determinat de

$$a_{11}x_1 + a_{12}x_2 \perp b_1$$

$$a_{21}x_1 + a_{22}x_2 \perp b_2$$

.....

$$a_{m1}x_1 + a_{m2}x_2 \perp b_m$$

și de $\min(\max)(c_1x_1 + c_2x_2)$.

Aflarea domeniului D se bazează pe separarea planului x_1Ox_2 în

regiunile determinate de dreptele care rezultă din

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 \perp b_1 \\ a_{21}x_1 + a_{22}x_2 \perp b_2 \\ \dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 \perp b_m \end{array},$$

și $\min(\max)(c_1x_1 + c_2x_2)$ sau altfel spus, domeniul D rezultă din intersectarea semispațiilor și a dreptelor conținute în

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 &\leq b_2 \\ &\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 &\leq b_m \end{aligned} \quad \text{și} \quad \min(\max)(c_1x_1 + c_2x_2). \quad \text{Domeniul } D \text{ este}$$

$$\begin{aligned} & a_{11}x_1 + a_{12}x_2 \leq b_1 \\ \text{determinat numai dacă sistemul } & a_{21}x_1 + a_{22}x_2 \leq b_2 \quad \text{și} \\ & \dots\dots\dots \\ & a_{m1}x_1 + a_{m2}x_2 \leq b_m \end{aligned}$$

$\min(\max)(c_1x_1 + c_2x_2)$ este compatibil. Punctele interioare și de frontieră ale lui D formează mulțimea soluțiilor posibile ale problemei. În această mulțime se caută soluția optimă (dacă există)..

Notăm cu $f = c_1x_1 + c_2x_2$ funcția de optimizat. Dacă interpretăm pe f ca un parametru, atunci $x_2 = -\frac{c_1}{c_2}x_1 + \frac{f}{c_2}$ reprezintă un fascicul de drepte paralele, cu ordonata la origine $\frac{f}{c_2}$. Pentru a construi

fasciculul $x_2 = -\frac{c_1}{c_2}x_1 + \frac{f}{c_2}$, se construiește dreapta ce trece prin origine

$x_2 = -\frac{c_1}{c_2}x_1$ care apoi se deplasează paralelă cu ea însăși. Presupunem

că pentru $m=4$ se obține domeniul D din figura 4.

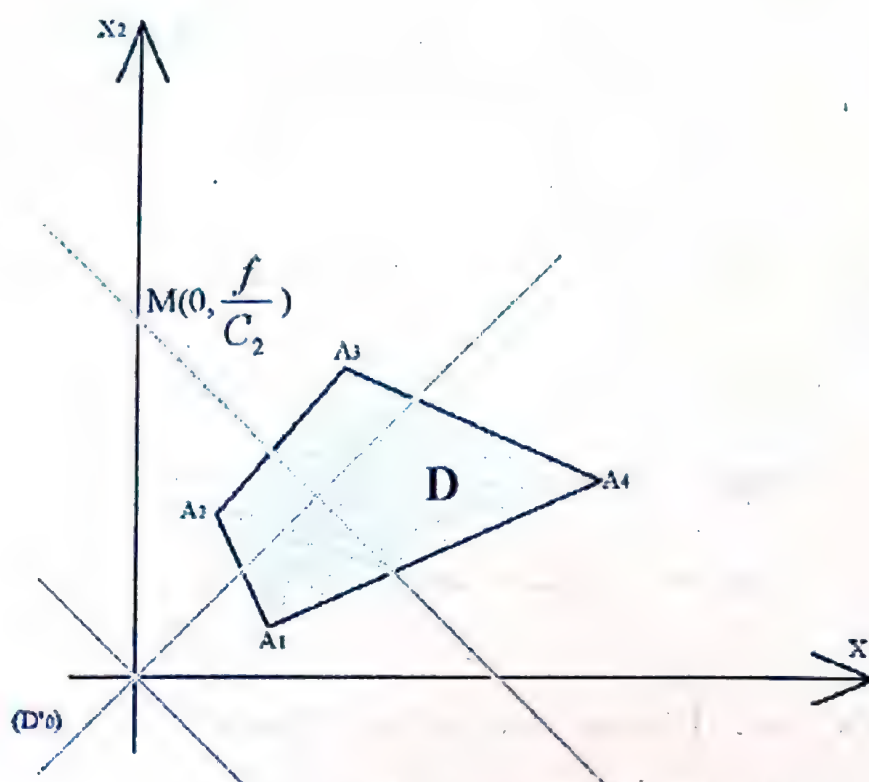


Figura 4. Punctele interioare și de frontieră ale lui D formează mulțimea soluțiilor posibile ale problemei date.

Toate punctele lui D (interioare și de pe frontieră) sunt soluții posibile ale problemei date. Funcție de valorile c_1 și c_2 dreapta $x_2 = -\frac{c_1}{c_2}x_1$ este de forma (D_0) sau (D'_0) . Pentru a determina soluția optimă, dreapta (D_0) sau (D'_0) se deplasează paralelă cu ea însăși așa încât $D \cap (D'_0) \neq \emptyset$. Sensul de deplasare depinde de valorile lui c_1 și c_2 , iar distanța cu care se deplasează depinde de D . Ne vom ocupa în

continuare numai de problema de maxim. Presupunem că $c_1 \neq 0$ și $c_2 \neq 0$. Apar următoarele patru cazuri:

a. $c_1 > 0, c_2 > 0$. Dreptele $x_2 = -\frac{c_1}{c_2}x_1 + \frac{f}{c_2}$ și $x_2 = -\frac{c_1}{c_2}x_1$ apar

ca în figura 5:

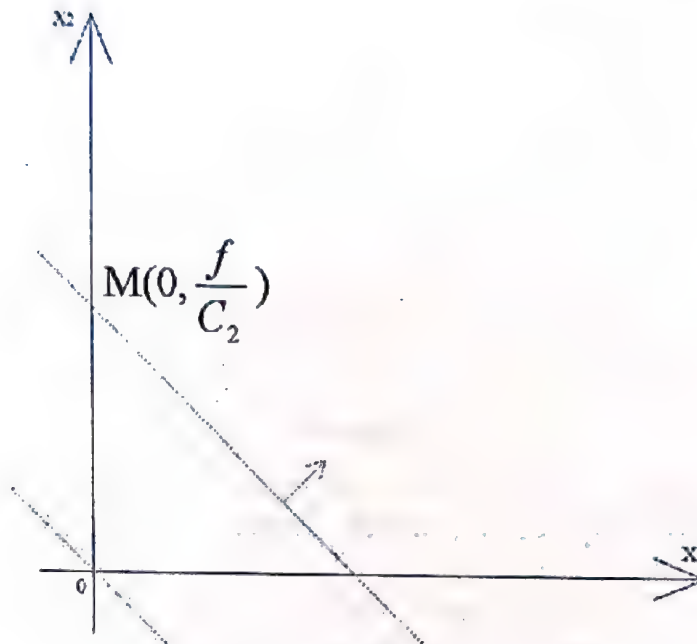


Figura 5.

și deci $\frac{f}{c_2}$ maxim implică f maxim (c_2 fiind o valoare constantă).

Înseamnă că valorile lui f cresc, când ordonata la origine crește. Sensul de deplasare a lui $x_1 = x_1^*, x_2 = x_2^*$ pentru atingerea valorii maxime se vede în figura 5.

b. $c_1 < 0, c_2 > 0$. Dreptele $x_2 = -\frac{c_1}{c_2}x_1 + \frac{f}{c_2}$ și $x_2 = -\frac{c_1}{c_2}x_1$ apar

ca în figura 6 :

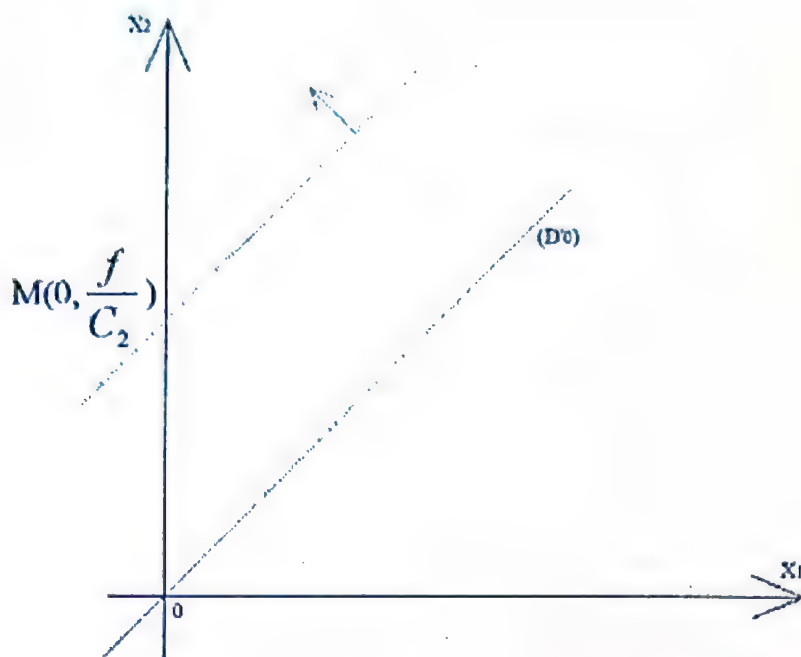


Figura 6

și deci $\frac{f}{c_2}$ maxim implică f maxim. Înseamnă că valorile lui f cresc, când ordonata la origine crește. Sensul de deplasare a lui (D'_0) pentru atingerea valorii maxime se vede în figură 6.

c. $c_1 < 0, c_2 < 0$. Dreptele $x_2 = -\frac{c_1}{c_2}x_1 + \frac{f}{c_2}$ și $x_2 = -\frac{c_1}{c_2}x_1$ apar

ca în figura 7:

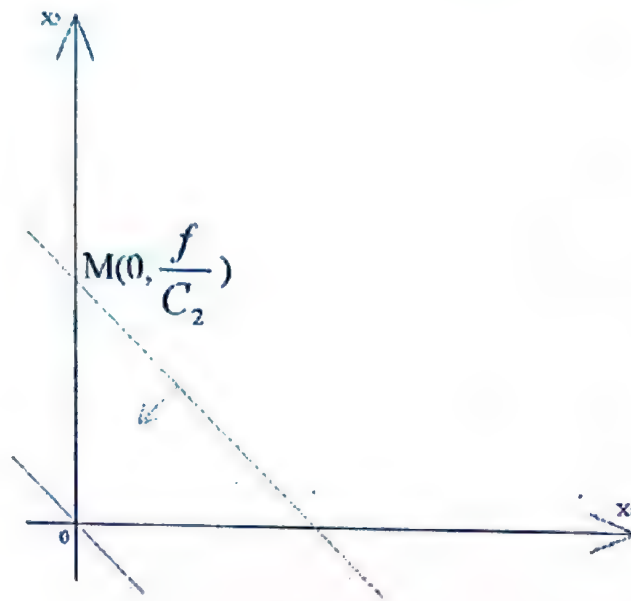


Figura 7.

și deci $\frac{f}{c_2}$ minim implică f maxim. Înseamnă că valorile lui f cresc, când ordonata la origine scade. Sensul de deplasare a lui (D_0) pentru atingerea valorii maxime se vede în figura 7.

d. $c_1 > 0, c_2 < 0$. Dreptele $x_2 = -\frac{c_1}{c_2}x_1 + \frac{f}{c_2}$ și $x_2 = -\frac{c_1}{c_2}x_1$ apar

ca în figura 8 :

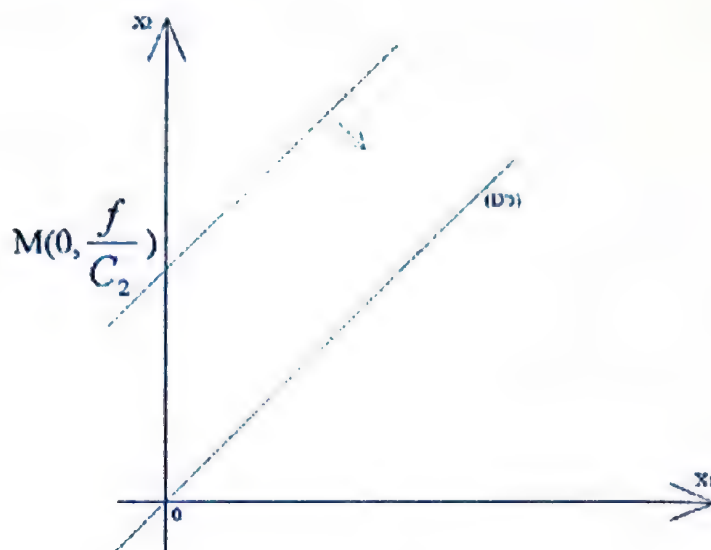


Figura 8

și deci $\frac{f}{c_2}$ minim implică f maxim. Înseamnă că valorile lui f cresc, când ordonata la origine scade. Sensul de deplasare a lui (D'_0) pentru atingerea valorii maxime se vede în figura 8.

$$\text{Pentru problema } \left\{ \begin{array}{l} \min(\max)(c_1 x_1 + c_2 x_2) \\ a_{11} x_1 + a_{12} x_2 \leq b_1 \\ a_{21} x_1 + a_{22} x_2 \leq b_2 \\ \dots\dots\dots \\ a_{m1} x_1 + a_{m2} x_2 \leq b_m \end{array} \right. \text{ de minim, sensurile în}$$

figura 8 se inversează în fiecare caz în parte.

Presupunem că pentru domeniul D din figura 4 avem cazul din figura 5. Atunci A_4 realizează maximul lui f , iar A_1 realizează minimul lui.

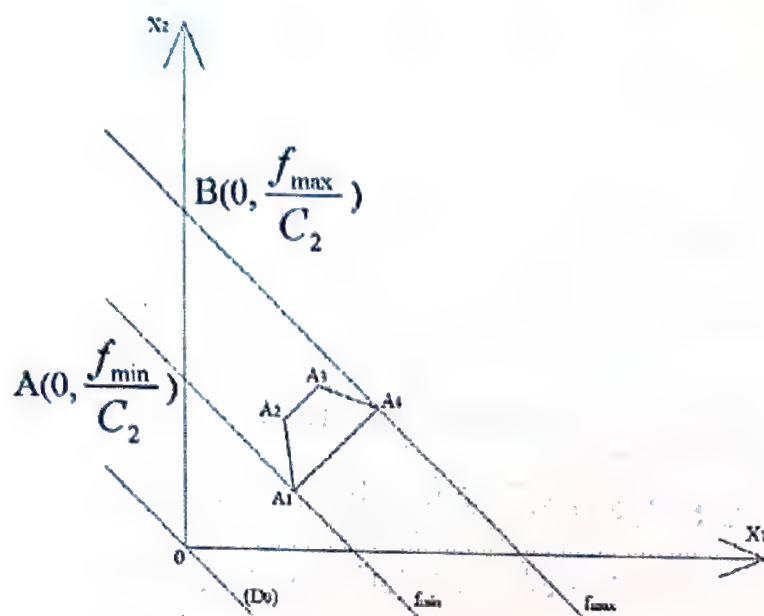


Figura 9. Soluția optimă a problemei de maxim este punctul $A_4(x_1^*, x_2^*)$ iar soluția optimă a problemei de minim este punctul $A_1(x_1', x_2')$.

$$f_{\max} = c_1 x_1^* + c_2 x_2^*$$

$$f_{\min} = c_1 x_1' + c_2 x_2'$$

unde $A_4(x_1^*, x_2^*)$ iar $A_1(x_1', x_2')$. Deci soluția optimă a problemei de maxim este

$x_1 = x_1^*, x_2 = x_2^*$, iar valoarea maximă a lui f este f_{\max} ; soluția optimă a problemei de minim este $x_1 = x_1', x_2 = x_2'$, iar valoarea minimă a lui f este f_{\min} .

Din figura de mai sus se observă că dacă se determină ordonata la origine OB sau OA, atunci valoarea f_{\max} , respectiv f_{\min} se obține prin înmulțirea cu c_2 a ordonatei la origine corespunzătoare.

Să urmărim rezolvarea următoarei probleme prin metoda geometrică:

Să se determine (x_1, x_2) așa încât

$$\begin{cases} \max(2x_1 + x_2) \\ x_1 - x_2 \leq 3 \\ 2x_1 - x_2 \leq 8, \\ -x_1 + 2x_2 \leq 6 \\ x_1, x_2 \geq 0 \end{cases}$$

Domeniul soluțiilor posibile se vede în figura 10:

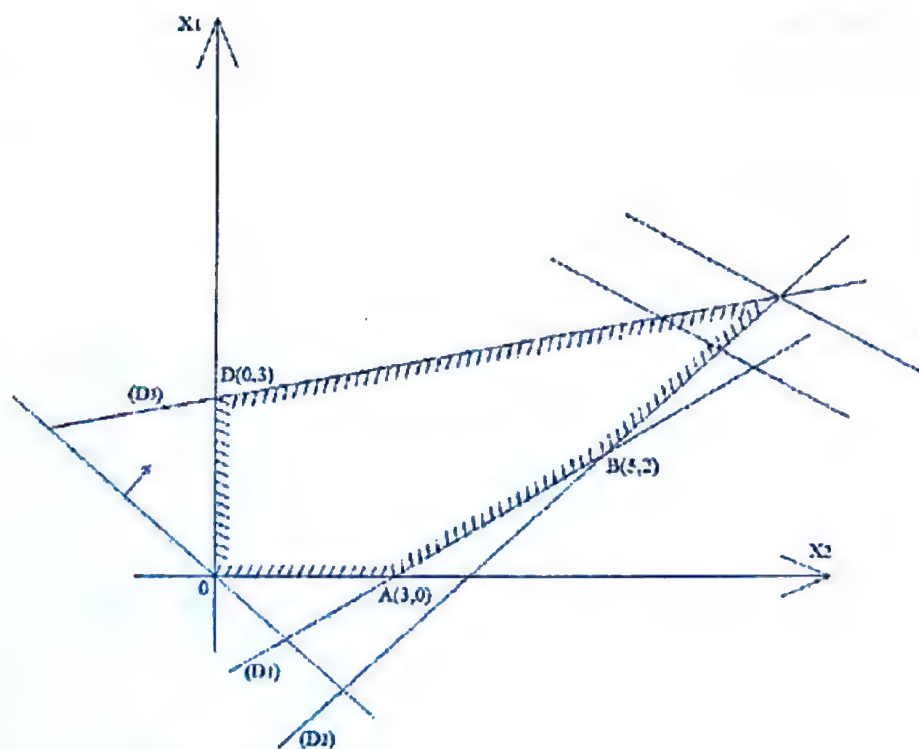


Figura 10. Maximul lui f se realizează în punctul C.

unde prin D_1, D_2, D_3 am notat dreptele generate de restricțiile problemei (în ordinea în care sunt date în enunțul problemei). Deplasarea dreptei

$$2x_1 + x_2 = f \text{ în sensul de creștere a ordonatei arată că } C\left(\frac{22}{3}, \frac{20}{3}\right)$$

realizează maximul lui f în condițiile date și anume:

$$f_{\max} = 2\frac{22}{3} + \frac{20}{3} = \frac{64}{3}; x_1^* = \frac{22}{3}, x_2^* = \frac{20}{3}.$$

În cazul când domeniul D definit de restricțiile problemei și dreapta (D_0) este paralelă cu dreapta A_3A_4 , problema de maxim are o infinitate de soluții optime – anume toate punctele de pe segmentul A_3A_4 - iar problema de minim are soluție optimă unică, realizată în A_1 așa cum se observă în figura 11.

Funcția de la optimizat f cu $c_1 \neq 0, c_2 = 0$ sau $c_1 = 0, c_2 \neq 0$ se studiază ca un aspect particular al cazurilor a-d.

În exemplele propuse spre a fi rezolvate ca exerciții se vor întâlni cazuri când problema de maxim nu are soluție optimă, iar problema de minim cu aceleași restricții și aceeași funcție de optimizat are soluție optimă, și invers.

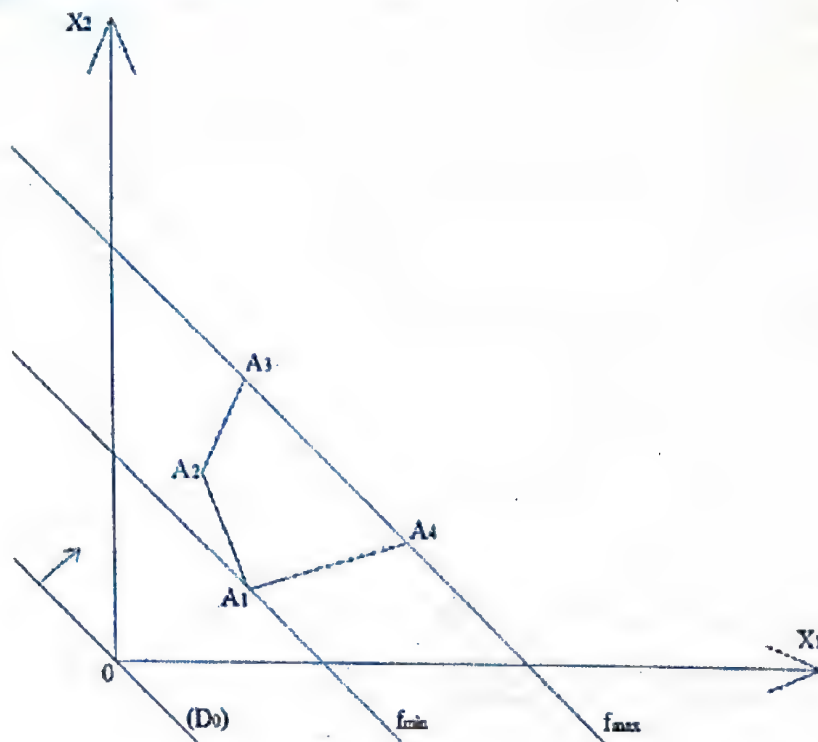


Figura 11. Problema de maxim are o infinitate de soluții optime – anume toate punctele de pe segmentul A_3A_4 , iar problema de minim are soluție optimă unică, realizată în punctul A_1 .

Generalizarea teoretică a rezolvării geometrice a unei probleme de programare liniară cu mai mult de două variabile este simplă, dar aplicarea ei practică este incomodă chiar pentru trei sau mai patru variabile și este imposibilă pentru mai mult de patru variabile.

3. Soluții

Dacă există m restricții cu n variabile se pot obține următoarele categorii de soluții:

- *Soluții posibile*, care definesc soluțiile ce se află în domeniul posibil Δ_p . Notând soluția cu φ_1 pentru soluțiile posibile se poate scrie: $\varphi \in \Delta_p$, ceea ce are loc când x satisface restricțiile $Ax = b$ și $x \geq 0$.
- *Soluțiile de bază* sunt acele soluții care se obțin făcând variabilele $m-n$, egale cu zero (pe rând) și rezolvând sistemul de restricții ca ecuații simultane.
- *Soluțiile posibile de bază* sunt soluțiile de bază din domeniul posibil.
- *Soluțiile optime* sunt denumite soluțiile posibile, care satisfac funcția obiectiv.
- *Soluția optimă de bază* este soluția posibilă de bază care satisface funcția obiectiv.
- *Soluții aparent optime* sunt acele soluții care asigură un optim local sau aparent al funcției obiectiv.
- *Soluții optime globale* sunt acele soluții care asigură optimul absolut al funcției obiectiv și care constituie principalul obiectiv al rezolvării problemelor de programare liniară.

Teoremele fundamentale ale programării liniare

Teorema 1

Dacă problema de programare liniară sub forma standard

$$\min(\max)[Z(x)] = c'x$$

$$Ax = b$$

$$x \geq 0$$

admite un program, soluție admisibilă, adică aceea care satisface atât restricțiile, cât și condițiile de semn impuse variabilelor, atunci ea are cel puțin un program (soluție admisibilă de bază).

Teorema 2

Mulțimea soluțiilor admisibile ale unei probleme de programare liniară este o mulțime convexă.

Presupunem că problema este dată în forma standard.

$$\min(\max)[Z(x)] = c'x$$

$$Ax = b$$

$$x \geq 0$$

Se consideră două soluții admisibile x_1, x_2 . Combinația lor convexă este:

$$\bar{x} = (1 - \lambda)x_1 + \lambda x_2, \lambda \in [0, 1]$$

$$1 - \lambda \geq 0, \lambda \geq 0, \bar{x} \geq 0$$

$$A\bar{x} = A[(1 - \lambda)x_1 + \lambda x_2] = A(1 - \lambda)x_1 + A\lambda x_2$$

dar

$$Ax_1 = b$$

$$Ax_2 = b$$

deci

$$b = b\lambda + (1 - \lambda)b$$

ceea ce dovedește afirmația.

Teorema 3

Dacă problema de programare liniară

$$\min(\max)[Z(x)] = c'x$$

$$Ax = b$$

$$x \geq 0$$

admite un program optim, atunci admite cel puțin un program optim de bază.

Teorema 4

Funcția obiectiv a unei probleme de programare liniară ia valoarea optimă într-un punct extremal al mulțimii convexe Δ a tuturor soluțiilor admisibile ale problemei.

CAPITOLUL II

MODELE TEHNICO - ECONOMICE DE OPTIMIZARE CE CONDUC LA O PROBLEMĂ DE PROGRAMARE LINIARĂ

a) Consumul lunar de materii prime

Considerăm un proces de producție în cursul căruia se realizează m tipuri de produse P_i , $i = \overline{1, m}$, pentru care se utilizează n tipuri de materii prime M_j , $j = \overline{1, n}$. Dintr-o unitate de materie primă M_j se pot produce a_{ij} unități de produs P_i . Se preconizează ca produsele de tipul P_i să fie realizate lunar în cantitățile b_i . Costul unei unități de materie primă M_j este C_j . Se cere a determina consumul lunar de materii prime, încât să se realizeze producția planificată, iar cheltuielile legate de materiile prime să fie minime.

Urmărim modelul matematic al acestui model economic.

Vom însemna cu x_j , $j = \overline{1, n}$, cantitățile de materie primă ce urmează a fi utilizate. Va trebui să scriem sub formă algebrică restricțiile economice. Aceste restricții sunt de trei feluri: restricții legate de planificarea producției, restricții impuse de sensul economic al variabilelor, și restricții de minimizare a cheltuielilor. Așa cum $a_{ij}x_j$ reprezintă cantitatea de produse de tipul P_i ce se obține dintr-o cantitate x_j de materie primă M_j , deducem că:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = \sum_{j=1}^n a_{ij}x_j$$

reprezintă cantitatea totală de produse de tipul P_i , ce se obține când materiile prime M_j se utilizează în cantitățile x_j , $j = \overline{1, n}$. Dar această cantitate nu poate fi mai mică decât b_i și deci:

$$\sum_{j=1}^n a_{ij}x_j \geq b_i, \quad i = \overline{1, m} \quad (1)$$

Cum x_j reprezintă mărimi economice rezultă:

$$x_j \geq 0, \quad j = \overline{1, n} \quad (2)$$

Cheltuielile totale obținute ca sumă a cheltuielilor unitare vor fi date de funcția:

$$f(x) = \sum_{j=1}^n c_j x_j; \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \quad (3)$$

care se cere a fi minimizată.

Din punct de vedere matematic, problema poate fi formulată astfel: să se găsească în mulțimea soluțiilor sistemului de inecuații (1) care satisfac condițiile (2), o soluție pentru care funcția f ia valoarea minimă.

Problema are sens dacă sistemul (1) are cel puțin o soluție. Inecuațiile (1) reprezintă restricțiile problemei, (2) condițiile de nenegativitate iar f poartă denumirea de funcție obiectiv.

b) Folosirea optimă a resurselor

Într-un proces de producție în care se pot realiza n tipuri de articole A_j , $j = \overline{1, n}$ sunt utilizate m tipuri de resurse R_i , $i = \overline{1, m}$, care sunt limitate de cantitățile b_i . Pentru producerea unei unități de articol de tipul A_j se consumă o cantitate a_{ij} de resursă de tipul R_i . Profiturile unitare pentru articolele A_j sunt c_j , $j = \overline{1, n}$. Să se determine cantitățile x_j de articol A_j , $j = \overline{1, n}$ care urmează a se realiza în procesul considerat, cu resurse limitate, astfel ca procesul să aibă drept rezultat profitul maxim.

Limitările resurselor conduc la:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}.$$

Necunoscutele x_j , $j = \overline{1, n}$ trebuie să satisfacă condițiile de nenegativitate:

$$x_j \geq 0, \quad j = \overline{1, n}.$$

Profitul total este dat de funcția liniară:

$$f(x) = \sum_{j=1}^n c_j x_j \text{ care trebuie să ia valoarea maximă.}$$

c) Problema amestecului optim

Să presupunem că pentru realizarea unui amestec se pot întrebuința n materii prime M_j , $j = \overline{1, n}$, disponibile în cantități nelimitate, ce conțin m substanțe S_i , $i = \overline{1, m}$.

Dintr-o unitate de materie primă M_j , trece în amestec o cantitate a_{ij} de substanță S_i . Substanța S_i trebuie să intre în amestec în cantitatea b_i . Costurile unitare ale materiilor prime M_j sunt c_j unități bănești. Să se determine cantitățile x_j , $j = \overline{1, n}$ de materie primă M_j necesare realizării amestecului cu compoziția prescrisă și la un cost total minim.

Restricțiile sunt și acum de trei feluri: restricțiile de compoziție a amestecului, restricțiile datorate sensului economic al variabilelor și restricția de minimizare a costului total.

Traducerea algebrică a primelor două categorii este:

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m}$$

$$x_j \geq 0, \quad j = \overline{1, n}.$$

Costul total al materiilor prime necesare pentru realizarea unei unități de amestec este dat de funcția:

$$f(x) = \sum_{j=1}^n c_j x_j$$

care se cere a fi minimizată.

d) Un program de producție și stocaj

În cursul a n luni trebuie produse $r_i, 1 \leq i \leq n$, unități dintr-o anumită categorie.

Un orar normal permite un volum de producție de $\bar{x}_i, 1 \leq i \leq m$, unități pe lună. Se poate prevedea o producție suplimentară de \bar{x}'_i unități pe lună. Costurile unitare de producție sunt c_i, c'_i lunar, respectiv în primul și al doilea caz. Costul unitar de stocaj pe lună este d . Se cere să se afle cantitățile $x_i, x'_i, s_i, 1 \leq i \leq m$, ce trebuie produse în orar normal, în ore suplimentare, respectiv stocate, astfel încât să fie respectate condițiile:

$$\sum_{k=1}^i (x_k + x'_k) - s_{i+1} = \sum_{k=1}^i r_k, 1 \leq i \leq n-1,$$

$$0 \leq x_i \leq \bar{x}_i, 1 \leq i \leq n$$

$$0 \leq x'_i \leq \bar{x}'_i, 1 \leq i \leq n,$$

$$s_i \geq 0, 1 \leq i \leq n$$

și să se obțină minimul cheltuielilor de producție și stocaj:

$$\min \left[\sum_{i=2}^n (c_i x_i + c'_i x'_i + d s_i) + c_1 x_1 + c'_1 x'_1 \right]$$

e) Utilizarea optimă a capacității mașinilor

O întreprindere produce mai multe produse care pot fi fabricate de aceeași mașină a cărei capacitate de producție este limitată; se cere un program de producție care să asigure utilizarea optimă a mașinilor.

Mai precis, uzina produce n produse distincte, care pot fi fabricate cu ajutorul a m mașini (sau secții de producție) care au capacități limitate. Notăm prin $a_{ij}, 1 \leq i \leq m, 1 \leq j \leq n$, procentul din capacitatea mașinii i pe perioada considerată necesar pentru producerea unei unități din produsul j , iar prin $x_j, 1 \leq j \leq n$, numărul unităților din produsul j fabricate în cursul perioadei. Avem restricții de capacitate de forma

$$\sum_{j=1}^n a_{ij} x_j \leq 1, 1 \leq i \leq m$$

și problema se completează adăugând

$$x_j \geq 0, 1 \leq j \leq n,$$

$$\max \sum_{j=1}^n c_j x_j$$

unde c_j sunt beneficiile unitare.

f) O problemă de investiții

Avem la dispoziție o sumă totală S care poate fi investită în diverse activități j , $1 \leq j \leq n$, fiecare producând un anumit beneficiu unitar $a_j, 1 \leq j \leq n$. Dacă $x_j, 1 \leq j \leq n$, este suma investită pentru activitatea j , problema este

$$\max \sum_{j=1}^n a_j x_j$$

$$\sum_{j=1}^n x_j = S$$

$$x_j \geq 0, 1 \leq j \leq n$$

Problema poate fi complicată dând în plus anumite reguli suplimentare în legătură cu posibilitatea de investiție, cu existența unui risc al investițiilor sau cu alte aspecte suplimentare care pot fi întâlnite în condițiile concrete.

g) Reducerea pierderilor la tăierea materialelor

Vom considera o problemă simplă, care apare în activitatea de tăiere a hârtiei la o fabrică de celuloză, care produce rulouri de hârtie de lățime dată, depinzând de caracteristicile mașinii. Aceste rulouri trebuie tăiate pentru a satisface comenzile beneficiarilor ceea ce generează pierderi; problema constă în minimizarea acestor pierderi.

Să notăm cu

$$a_i, 1 \leq i \leq m,$$

$$b_i, 1 \leq i \leq m$$

respectiv lățimea și lungimea celor m rulouri comandate de beneficiari, iar prin l lățimea ruloului standard produs de mașină.

Se determină toate combinațiile posibile k , $1 \leq k \leq N$, în care poate fi tăiat ruloul standard pentru a obține

$$d_{ik}, 1 \leq i \leq m, 1 \leq k \leq N, \text{ rulouri de lățime } a_i \text{ (evident } 0 \leq d_{ik} \leq \left\lfloor \frac{l}{a_i} \right\rfloor \text{),}$$

unde $[x]$ este partea întragă a numărului x .

Dacă se notează cu $x_k, 1 \leq k \leq N$, lungimea ruloului standard în cazul în care se aplică tăierea de tipul k , avem condițiile

$$\sum_{k=1}^N d_{ik} x_k \geq b_i, 1 \leq i \leq m,$$

$$x_k \geq 0, 1 \leq k \leq N,$$

sau în forma standard:

$$\sum d_{ik} x_k - x_i^0 = b_i, 1 \leq i \leq m,$$

$$x_k \geq 0, 1 \leq k \leq N,$$

$$x_i^0 \geq 0, 1 \leq i \leq m,$$

Dacă notăm cu $c_k, 1 \leq k \leq N$, pierderea prin tăiere în cazul procedeului de tip k , pierderea totală care trebuie minimizată va fi

$$\min \left(\sum_{k=1}^N c_k x_k + \sum_{i=1}^m a_i x_i^0 \right).$$

Formularea de mai sus presupune existența unei singure mașini. Cazul mai multor mașini este mai dificil.

h) O tipografie care lucrează 25 de ore pe săptămână tipărește trei tipuri de cărți. Beneficiul obținut în urma vânzării acestor cărți este de 40 lei/buc pentru primul articol, respectiv 120 lei/buc pentru al doilea articol și 30 lei/buc pentru ultimul articol. Într-o oră mașina poate realiza 50 buc din primul articol sau 25 buc din al doilea articol sau 75 buc din al treilea articol. Cererea săptămânală nu depășește 1000 buc din primul articol, 500 buc din al doilea articol, 1500 buc din al treilea articol. Cum trebuie repartizată producția celor trei articole pentru ca tipografia să-și asigure un beneficiu maxim:

Răspuns: Fie x_1, x_2, x_3 cantitățile din cele trei articole ce trebuie tipărite. Atunci avem de rezolvat problema:

$$\max f = 40x_1 + 120x_2 + 30x_3$$

$$\begin{cases} x_1 \leq 1000 \\ x_2 \leq 500 \\ x_3 \leq 1500 \\ \frac{1}{50}x_1 + \frac{1}{25}x_2 + \frac{1}{75}x_3 \leq 25 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

($\frac{1}{50} \cdot x_1$ reprezintă numărul de ore pe săptămână în care tipografia produce cantitatea x_1 din primul articol).

i) O fabrică de zahăr trebuie să producă între 1 noiembrie - 28 februarie. Livrările de zahăr se fac lunar, după contract: în noiembrie 20.000 t, în decembrie 30.000 t, în ianuarie 50.000 t și în februarie 40.000 t. Producția excedentară a unei luni poate fi livrată luna următoare, suportând cheltuielile de depozitare de 2.000 lei/tonă pe lună. Capacitatea lunară de producție a fabricii este: 55.000 t în noiembrie, 40.000 t în decembrie, 25.000 t în ianuarie, 50.000 t în februarie. Prețurile de cost sunt: 140.000 lei/tonă în noiembrie, 160.000 lei/tonă în decembrie, 150.000 lei/tonă în ianuarie, 170.000 lei/tonă în februarie.

Să se stabilească nivelul de producție lunar astfel încât contractele să fie satisfăcute cu cheltuieli minime.

Răspuns. Vom presupune că producția se termină fără stoc. Fie x_1, x_2, x_3, x_4 cantitățile de zahăr ce trebuiesc produse respectiv în lunile: noiembrie, decembrie, ianuarie, februarie. Avem:

$$x_i > 0 \quad i = \overline{1,4}$$

$$\begin{cases} x_1 \leq 55000 \\ x_2 \leq 40000 \\ x_3 \leq 25000 \\ x_4 \leq 50000 \\ x_1 \geq 20000 \\ x_2 + (x_1 - 20000) \geq 30000 \\ x_3 + (x_1 + x_2 - 50000) \geq 50000 \\ x_4 + (x_1 + x_2 + x_3 - 100.000) = 40.000 \end{cases}$$

$((x_1 - 20000))$ - reprezintă excedentul din noiembrie.)

Considerând cheltuielile de producție și cele de depozitare în mii lei funcția obiectiv devine:

$$f(x) = 140x_1 + 160x_2 + 150x_3 + 170x_4 + 2(x_1 - 20000) + 2(x_1 + x_2 - 50000) + 2(x_1 + x_2 + x_3 - 100000)$$

Deci avem de calculat minimul funcției:

$$f(x) = 146x_1 + 164x_2 + 152x_3 + 170x_4 - 340000.$$

j)) Un laborator de informatică dispune de cablu utp cu lungimea de 5m, din care trebuie să taie cel puțin 35 cabluri în lungime de 2m, 25 cabluri de 2,5m și 10 cabluri de 3m lungime. Cum trebuie procedat astfel ca să se realizeze consumuri minime de material?

Răspuns: Există patru variante de a tăia un cablu de 5m lungime în bucăți cu lungimile specificate:

$$v_1: \quad 5 = 2 + 2 + 1$$

$$v_2: \quad 5 = 2,5 + 2,5$$

$$v_3: \quad 5 = 3 + 2$$

$$v_4: \quad 5 = 2 + 2,5 + 0,5$$

Observăm că la prima variantă se pierde 1m de cablu, la cea de-a patra 0,5m iar la variantele a II-a și a III-a nimic. Notând cu x_1, x_2, x_3, x_4 numărul cablurilor de 5m care se taie respectiv după variantele v_1, v_2, v_3, v_4 vom avea:

$$\min f = x_1 + 0,5x_4.$$

$$\begin{cases} 2x_1 + x_3 + x_4 \geq 35 \\ 2x_2 + x_4 \geq 25 \\ x_3 \geq 10 \\ x_i > 0 \quad i = \overline{1,4} \end{cases}$$

Vom obține $x_1 = 0, x_2 = 5, x_3 = 10, x_4 = 15$

j) Fiecare animal dintr-o grădină zoo are nevoie de o cantitate minimă de principii nutritive pe zi care depinde de specie, vârstă, scop urmărit în alimentație.

Principiile nutritive se află în diferite proporții în produsele ce compun rația zilnică a fiecărui animal. Folosind datele din tabelul de mai jos să se determine cantitatea x din alimentul A_1 și cantitatea y din alimentul A_2 exprimate în kilograme, ce trebuie să intre în compoziția rației zilnice a unui animal astfel încât costul ei să fie minim.

| Denumirea principiilor nutritive | Conținutul în principii nutritive al alimentelor (kg) | | Cantitățile minime prescrise (kg) |
|--|---|-------|---|
| | A_1 | A_2 | |
| P1 | 0,1 | 0 | 0,4 |
| P2 | 0 | 0,1 | 0,6 |
| P3 | 0,1 | 0,2 | 2 |
| P4 | 0,2 | 0,1 | 1,7 |
| Costul (lei/kg) | 240 | 80 | |

Indicație: Problema se transcrie matematic astfel:

$$\begin{cases} 0,1x \geq 0,4 \\ 0,1y \geq 0,6 \\ 0,1x + 0,2y \geq 2 \\ 0,2x + 0,1y \geq 1,7 \\ x, y \geq 0 \\ \min f = 240x + 80y \end{cases}$$

CAPITOLUL III

ALGORITMUL SIMPLEX

Pentru rezolvarea problemelor de programare liniară s-a impus *algoritmul simplex* datorat lui G. B. Dantzig (1951). Această metodă ne permite să explorăm în mod sistematic mulțimea programelor de bază ale unei probleme de programare liniară în forma standard prin trecerea de la un program de bază la alt program de bază “vecin” care este “cel puțin la fel de bun” ca cel precedent. Metoda furnizează de asemenea criterii pentru punerea în evidență a situației când problema are optim infinit precum și a cazului în care mulțimea programelor este vidă.

Algoritmul Simplex se poate prezenta pornindu-se de la forma standard a problemei de programare liniară, în modul următor:

$$\max[Z(x)] = cx$$

$$Ax = b$$

$$x \geq 0$$

alegând o bază oarecare B , vom scrie:

$$\max[Z(x)] = C^B X^B + C^S X^S$$

unde: X^B este vector coloană al variabilelor de bază; X^S este vector coloană al variabilelor secundare; C^B este vector coloană al costurilor variabilelor de bază din ecuația obiectiv, iar C^S este vectorul coloană al costurilor variabilelor secundare, din ecuația obiectiv.

Astfel, scriem în continuare:

$$(B, S) \begin{pmatrix} X^B \\ X^S \end{pmatrix} = b$$

$$\begin{pmatrix} X^B \\ X^S \end{pmatrix} \geq 0$$

sau

$$\max[Z(x)] = C^B X^B + C^S X^S$$

$$BX^B + SX^S = b$$

$$X \geq 0; (X^B \geq 0; X^S \geq 0)$$

Dacă înmulțim sistemul de restricție cu B^{-1} , matricea inversă a coeficienților variabili de bază din restricții, se obține pentru $AX = b$:

$$X^B = B^{-1}b - B^{-1}SX^S$$

iar ecuația va fi de forma:

$$\max[Z(x)] = C^B (B^{-1}b - B^{-1}SX^S) + C^S X^S$$

$$X^B \geq 0; X^S \geq 0.$$

Dacă introducem notațiile:

$$\bar{X}^B = B^{-1}b \text{ și } X_j^B = B^{-1}a_j$$

unde a_j reprezintă coloana j din matricea A corespunzătoare variabilelor secundare ($j \in J_s$) și unde J_s este mulțimea $\{j : a_j \in S\}$, atunci se obține egalitatea din relația $X^B = B^{-1}b - B^{-1}SX^S$:

$$X^B = \bar{X}^B - \sum_{j \in J_s} y_j^B x_j$$

sau

$$X_i^B = \bar{X}_i^{-B} - \sum_{j \in J_B} y_j^B x_j, (i \in J_B)$$

unde $J_B = \{i : a_i \in B\}$.

Se exprimă funcția obiectiv, cu ajutorul variabilelor secundare.

X^S .

$$\begin{aligned} Z = c'x &= \sum_{j=1}^m c_j x_j = \sum_{i \in J_B} c_i x_i + \sum_{j \in J_S} c_j x_j = \sum_{i \in J_B} c_i (x_i^B - \sum_{j \in J_B} y_{ij}^B x_j) + \sum_{j \in J_S} c_j x_j \\ &= \sum_{i \in J_B} c_i x_i^B - \sum_{j \in J_B} (\sum_{i \in J_B} c_i y_{ij}^B - c_j) x_j \end{aligned}$$

Notând:

$$Z^B = \sum_{i \in J_B} c_i x_i^B = c_B' x^B$$

$$Z_j^B = \sum_{i \in J_B} c_i y_{ij}^B = c_B' y_j^B = \sum_{i \in J_B} c_i a_{ij}.$$

Obținem relația:

$$Z = Z^B - \sum_{j \in J_B} (Z_j^B - c_j) x_j.$$

CRITERIUL OPTIM

Considerăm cazul minimizării funcției obiectiv. Dacă $Z_j^B - c_j \leq 0$ pentru orice $j \in J_S$ atunci soluția admisibilă corespunzătoare bazei B ($X^B = B^{-1}b, X^S = 0$) este optimă.

Adina Butnărașu

Demonstrație: Fie Y valoarea funcției obiectiv corespunzătoare soluției admisibile de bază, știind că $x_j \geq 0$, rezultă că:

$$\sum_{j \in J_B} (Z_j^B - c_j) x_j \leq 0,$$

ceea ce demonstrează afirmația de mai sus.

CRITERIUL DE INTRARE ÎN BAZĂ

Dacă există $k \in J_S$ astfel încât $Z_k - c_k > 0$, atunci soluția admisibilă pentru baza B nu este optimă și poate fi îmbunătățită dacă x_k ia valori pozitive.

Demonstrație: Dacă valoarea variabilei secundare crește până la valoarea $x_k > 0$, se obține pentru funcția obiectiv:

$$Z^0 = Z^B - \sum (z_k - c_k) x_k^0 < Z^B$$

ceea ce reprezintă un program mai bun (funcția obiectiv luând o valoare mai mică). Se constată că odată cu creșterea valorii lui x_k , se modifică și valorile variabilelor de bază, conform relației

$$X_i^B = x_i - \sum_{j \in J_B} y_{ik}^B x_k ; i \in J_B.$$

CRITERIUL OPTIM INFINIT

Dacă există $k \in J_s$ astfel încât $Z_k - c_k > 0$ și toți $y_{ik} < 0$ pentru $i \in J_B$, atunci problema are un optim la infinit. În adevăr, dacă $y_{ik} \leq 0$, atunci din relația $Z^B = \sum c_i \bar{x}^B = c_B^T \bar{x}^B$ rezultă că variabila x_k , oricât ar crește, nu determină ca vreuna din variabilele de bază să devină negativă.

În acest caz funcția obiectiv devine:

$$Z = Z^B - (z_k - c_k)x_k \rightarrow -\infty$$

$$Z = Z^B - (z_k - c_k)x_k \rightarrow +\infty$$

CRITERIUL DE IEȘIRE DIN BAZĂ

Dacă $Z_k - c_k > 0$ și $y_{ik} > 0$, atunci valoarea variabilei x_k poate crește până atinge valoarea ϕ_1 .

$$\phi_1 = \frac{\bar{x}_1}{y_{1k}} = \min_{i \in J_B} \left(\frac{\bar{x}_i}{y_{ik}} \right), y_{ik} > 0$$

caz în care se va obține un nou program de bază, asociat bazei \bar{B} dedusă din baza B prin înlocuirea vectorului a_k cu vectorul a_1 .

Dacă $y_{ik} > 0$, în acest caz $x_i = \bar{x}_i - y_{ik}x_k \geq 0$, pentru toți $i \in J_B$, este necesar ca:

$$x_k \leq \frac{x_i}{y_{ik}} \text{ pentru toți } i \in J_B, y_{ik} > 0.$$

Dacă presupunem că valoarea variabilei x_k poate crește numai până la valoarea arătată de expresia ϕ_1 și că acest minim este atins pentru indicele $1 \in J_B$, la această valoare x_1 se anulează și deci obținem un nou program de bază în care variabilele de bază sunt x_k și x_i ($i \in J_B - \{k\}$), adică un program corespunzător bazei B obținute prin înlocuirea factorului a_1

cu a_k . Deci x_k nu poate crește mai mult decât valoarea ϕ_1 , căci altfel ar deveni negativă valoarea variabilei x_1 . Valoarea funcției obiectiv scade cu cantitatea $\frac{x_1}{y_{1k}}(z_k - c_k)$.

În practică se folosește un criteriu mai simplu: se alege acel indice k pentru care valoarea lui $Z_j - c_j$ este maximă, adică:

$$z_k - c_k = \max(z_j - c_j).$$

Modul de aplicare a algoritmului Simplex la rezolvarea problemelor de programare liniară

Să considerăm cazul minimizării funcției $Z(x)$. Urmărim etapele:

a) Se determină baza inițială B . Se calculează apoi, pentru această

bază mărimile: $\bar{x}_i, \bar{z}^B, \bar{z}_j^B - c_j, y_{ij}^B (a_{ij})$.

b) Se analizează diferențele $Z_j - c_j$ (criteriul de intrare în bază).

- dacă toate diferențele $Z_j - c_j \leq 0$ în cazul minimizării, programul este optim;
- dacă există cel puțin un indice $j \in J_B$ astfel ca $Z_j - c_j > 0$, atunci se determină $k \in J_B$ pentru care $Z_k - c_k = \max |z_j - c_j| > 0$.

c) Se stabilește vectorul ce urmează a ieși din bază (criteriul de ieșire din bază);

- dacă toți $y_{ik} \leq 0$ (sau $a_{ik} \leq 0$), problema are optim infinit;
- dacă există $y_{ik} > 0$ sau $a_{ik} > 0$, se determină indicele l astfel încât să fie satisfăcută relația:

$$\phi_l = \frac{\bar{x}_l}{y_{lk}} = \min_{i \in J_B} \left| \frac{\bar{x}_i}{x_{lk}} \right| > 0 \text{ sau}$$

$$\phi_1 = \frac{\bar{x}_1}{a_{1k}} = \min_{i \in J_B} \left| \frac{\bar{x}_1}{x_{1k}} \right| > 0.$$

d) Se înlocuiește în baza inițială B vectorul a_1 cu vectorul a_k , și se obține o nouă bază \bar{B} , transformându-se și mărimile $\bar{x}, Z^B, z_j - c_j, y_{ij}^B$, după relațiile:

$$\bar{x}_k^{\bar{B}} = \frac{x_1}{y_{1k}}, \quad y_{kj}^{\bar{B}} = \frac{y_{1j}}{y_{1k}}$$

$$\bar{x}_i^{\bar{B}} = x_i - y_{ik} \frac{\bar{x}_1}{y_{1k}}; \quad y_{ij}^{\bar{B}} = y_{ij} - y_{ik} \frac{y_{1j}}{y_{1k}}, \quad i \neq 1$$

$$Z^{\bar{B}} = Z^B - (z_k - c_k) \frac{\bar{x}_1}{y_{1k}};$$

$$z_j^{\bar{B}} - c_j = (z_j - c_j) - (z_k - c_k) \frac{y_{1j}}{y_{1k}}$$

se reia aplicarea algoritmului de la pasul b).

Elementul situat la intersecția rândului 1 cu coloana k se numește pivot.

În cazul problemelor prin care se cere maximizarea funcției $Z(x)$, se modifică numai criteriul de intrare în bază, astfel:

Etapă (a) este identică ca în cazul minimizării; b). Dacă există cel puțin un indice $j \in J_S$ astfel încât $Z_j - c_j < 0$, se determină k astfel încât $Z_k - c_k = \min(z_j - c_j)$ sau $Z_k - c_k = \max|z_j - c_j| < 0$.

Pentru aplicarea algoritmului Simplex se trec într-un tabel numit Simplex toate elementele necesare.

Dacă pentru problemele de minim există diferențe

$$\Delta_{z_j} = z_j - c_j > 0$$

iar pentru cele de maximum $\Delta_{z_j} = z_j - c_j < 0$

atunci soluția $X^B = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m, 0, \dots, 0)$ nu este optimă și va intra în bază vectorul a_k , care se determină cu ajutorul criteriului de intrare

$$\Delta_{z_k} = \max |z_j - c_j|.$$

Vectorul a_k va intra în bază în locul vectorului a_1 , determinat prin criteriul de ieșire din bază:

$$\phi_1 = \min_{i \in J_B} \left(\frac{\bar{x}_1}{y_{1k}} \right) = \frac{\bar{x}_1}{y_{1k}} = \frac{\bar{x}_1}{\alpha_{1k}}; y_{1k} > 0.$$

În tabelul Simplex s-a luat inițial $y_{ij} = \alpha_{ij}$ și drept urmare pivot va fi elementul α_{1k} .

După aceasta se aplică relațiile cu ajutorul cărora se calculează elementele tabelului Simplex corespunzător bazei \bar{B} . Toate calculele necesare unei noi baze constituie o iterație a algoritmului Simplex.

Observație: Trebuie făcută precizarea că rezolvarea problemelor se face în Q și nu în Z . Există și algoritmi de rezolvare în numere

Adina Butnărașu

întregi, dar scopul lucrării este de a prezenta metoda și modul de modelare al unor probleme de optimizare.

Deoarece algoritmi de optimizare în Z sunt mai complicați, s-a preferat prezentarea problemei clasice.

Rezolvarea problemelor prin algoritmul simplex

Exemplul 1.

$$\begin{cases} \max[z(x)] = 10x_1 + 15x_2 \\ 2x_1 + 4x_2 + x_3 = 40 \\ 6x_1 + 2x_2 + x_4 \leq 60 \\ x_j \geq 0 (j = 1, 2, 3, 4) \end{cases}$$

Aducerea problemei la forma standard: introducerea variabilelor de compensare x_3, x_4 .

Matricea restricțiilor este:

$$A = \begin{pmatrix} 2 & 4 & 1 & 0 \\ 6 & 2 & 0 & 1 \end{pmatrix} \text{ și conține programul de bază } B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Soluția admisibilă inițială are următoarea structură:

$$\bar{X} = (0, 0, 40, 60)$$

iar funcția obiectiv

$$\bar{Z}^B(x) = 0$$

| | B | C_B | \bar{X}^B | C_j | 10 | 15 | 0 | 0 | θ_i |
|-------|------------------------|-------|-------------|-------|----------|----------|-------|-------|---------------------|
| | | | | a_j | a_1 | a_2 | a_3 | a_4 | |
| T_0 | a_3 | 0 | 40 | | 2 | 4 | 1 | 0 | $\frac{40}{4} = 10$ |
| | a_4 | 0 | 60 | | 6 | 2 | 0 | 1 | $\frac{60}{2} = 30$ |
| | $z_j = C_{B_i} a_{ij}$ | | 0 | | 0 | 0 | 0 | 0 | - |
| | $z_j = z_j - c_j$ | | - | | 0-10=-10 | 0-15=-15 | 0 | 0 | - |
| | | | | | | | | | |

Pentru tabelul T_0 se calculează următoarele:

$$\bar{Z}^B = \sum_{i \in J_B} c_{B_i} \bar{x}_i = 0 \times 40 + 0 \times 60 = 0$$

$$z_1^B = \sum_{i \in J_B} c_{B_i} a_{i_1} = 0 \times 2 + 0 \times 6 = 0$$

$$z_2^B = \sum_{i \in J_B} c_{B_i} a_{i_2} = 0 \times 4 + 0 \times 2 = 0$$

$$\Delta z_j = z_j - c_j$$

$$\Delta z_1 = z_1 - c_1 = 0 - 10 = -10$$

$$\Delta z_2 = z_2 - c_2 = 0 - 15 = -15$$

$$\Delta z_3 = 0$$

$$\Delta z_4 = 0$$

Concluzia este că această soluție nu este optimă, deoarece există diferența $\Delta z_j < 0$ și de aceea se introduce în bază $\Delta z_k = \max |z_k - c_k|$ și pentru că această condiție se îndeplinește Δz_k se introduce în bază vectorul a_2 .

Pentru a determina ce vector urmează a fi scos din bază calculează rapoartele $\theta_i = \frac{x_i}{a_{i2}}$ (pentru $a_{i2} > 0$), coloana vectorilor, a_{i2} corespunzător vectorului a_2 care intră în bază. Se obține:

$$\theta_1 = \frac{x_3}{a_{12}} = \frac{40}{4} = 10;$$

$$\theta_2 = \frac{x_4}{a_{22}} = \frac{60}{2} = 30$$

Raportul minim $\theta = 10$ corespunde factorului a_3 , care iese din bază și în locul lui va intra a_2 , pivotal va fi 4.

Se dispune acum la o nouă bază $\bar{B} = (a_2, a_4)$.

| | B | C_B | \bar{X} | C_j | 10 | 15 | 0 | 0 | θ_i |
|-------|------------------------|-------|-----------|-------|-------|-------|-------|-------|------------|
| | | | | a_j | a_1 | a_2 | a_3 | a_4 | |
| T_1 | a_2 | 15 | 10 | | 1/2 | 1 | 1/4 | 0 | 20 |
| | a_4 | 0 | 40 | | 5 | 0 | -1/2 | 1 | 8 |
| | $z_j = C_{B_i} a_{ij}$ | | 150 | | 15/2 | 15 | 15/4 | 0 | - |
| | $z_j = z_j - c_j$ | | - | | -5/2 | 0 | 15/4 | 0 | - |
| | | | | | | | | | |

Pentru tabelul T_1 se calculează următoarele:

- a) Pentru determinarea vectorilor din primul rând $i=1$, se împart elementele din tabelul T_0 la valoarea pivotului a_{12} conform relației:

$$x_i^B = \bar{x}_i - \sum_{j \in J_B} y_{ik}^B x_k; i \in J_B$$

deci:

$$\bar{x}_2^B = \frac{\bar{x}_3}{a_{12}} = \frac{40}{4} = 10$$

$$y_{11}^B = \frac{a_{11}}{a_{12}} = \frac{2}{4} = \frac{1}{2}$$

$$y_{12}^B = \frac{a_{12}}{a_{12}} = 1$$

$$y_{13}^B = \frac{a_{13}}{a_{12}} = \frac{1}{4}$$

- b) Pentru determinarea celui de-al doilea rând $i=2$ din T_1 se folosesc relațiile:

$$\bar{x}^{\bar{B}} = x_4 - \frac{\bar{x}_3}{a_{12}} a_{22} = 60 - \frac{40}{4} \cdot 2 = 40$$

$$y_{21}^{\bar{B}} = a_{21} - \frac{a_{11}}{a_{12}} a_{22} = 6 - \frac{2}{4} \cdot 2 = 5$$

$$y_{23}^{\bar{B}} = a_{23} - \frac{a_{13}}{a_{12}} a_{22} = 0 - \frac{1}{4} \cdot 2 = -\frac{1}{2}$$

- c) indicatorii \bar{z}^B și $z_j^{\bar{B}} - c_j$ se calculează cu relațiile:

$$\bar{Z}^B = \sum_{i \in J_B} c_i x_i^{\bar{B}} = 15 \cdot 10 + 0 \cdot 40 = 150$$

sau

$$\bar{Z}^{\bar{B}} = \bar{Z}^B - (z_2 - c_2) \frac{\bar{x}_3}{a_{12}} = 0 - (-15) \cdot \frac{40}{4} = 150$$

$$z_1^{\bar{B}} = \sum_{i \in J_B} c_i a_{i1} = \frac{1}{2} \cdot 15 + 0 \cdot 5 = \frac{15}{2}$$

$$z_2^{\bar{B}} = \sum_{i \in J_B} c_i a_{i2} = 15 \cdot 1 + 0 \cdot 0 = 15$$

$$z_3^{\bar{B}} = \sum_{i \in J_B} c_i a_{i3} = 15 \cdot \frac{1}{4} + 0 \cdot \left(-\frac{1}{2}\right) = \frac{15}{4}$$

$$\Delta z_1^{\bar{B}} = z_1^{\bar{B}} - c_1 = \frac{15}{2} - 10 = -\frac{5}{2}$$

$$\Delta z_2^{\bar{B}} = z_2^{\bar{B}} - c_2 = 15 - 15 = 0$$

$$\Delta z_3^{\bar{B}} = z_3^{\bar{B}} - c_3 = \frac{15}{4} - 0 = \frac{15}{4}$$

sau

$$\Delta z_1^{\bar{B}} = \Delta z_1^B - \frac{a_{11}}{a_{12}} \Delta z_2^B = -10 + \frac{1}{2} \cdot 15 = -\frac{5}{2}$$

$$\Delta z_2^{\bar{B}} = \Delta z_2^B - \frac{a_{12}}{a_{12}} \Delta z_2^B = -15 + 1 \cdot 15 = 0$$

$$\Delta z_1^{\bar{B}} = \Delta z_3^B - \frac{a_{13}}{a_{12}} \Delta z_2^B = 0 + \frac{1}{4} \cdot 15 = \frac{15}{4}$$

Analizând valorile $\Delta z_i^{\bar{B}}$ observăm că $\Delta z_1^{\bar{B}} = -\frac{5}{2}$ ceea ce determină schimbarea bazei. Vectorul care intră va fi a_1 . Să determinăm vectorul care va ieși, determinând rapoartele:

$$\theta_i = \frac{x_i}{a_{i1}} (a_{i1} > 0)$$

$$\theta_1 = \frac{x_2}{a_{11}} = \frac{10}{\frac{1}{2}} = 20$$

$$\theta_2 = \frac{x_4}{a_{21}} = \frac{40}{5} = 8$$

Valoarea minimă pentru θ_i este $\theta_2 = 8$ ce corespunde vectorului a_4 , care se va scoate din baza nouă; $\bar{B} = (a_1, a_2)$ și trece la elaborarea tabelului T_2 . Pivotal va fi 5.

| | B | C_B | \bar{X} | C_j | 10 | 15 | 0 | 0 | θ_i |
|-------|------------------------|-------|-----------|-------|-------|-------|-------|-------|------------|
| | | | | a_j | a_1 | a_2 | a_3 | a_4 | |
| T_2 | a_2 | 15 | 6 | | 0 | 1 | 3/10 | -1/10 | |
| | a_1 | 10 | 8 | | 1 | 0 | -1/10 | 1/5 | |
| | $z_j = C_{B_i} a_{ij}$ | | 170 | | 10 | 15 | 7/2 | 1/2 | - |
| | $z_j = z_j - c_j$ | | - | | 0 | 0 | 7/2 | 1/2 | - |
| | | | | | | | | | |

- a) Pentru determinarea vectorilor din al doilea rând $i=2$ din tabelul T_2 , se împart elementele din tabelul T_1 , la valoarea pivotului $a_{21} = 5$ conform relației:

$$x_1^{\bar{B}} = \frac{\bar{x}_1}{y_{1k}} = \frac{\bar{x}_4}{y_{21}} = \frac{40}{5} = 8$$

$$y_{21}^{\bar{B}} = \frac{a_{21}}{a_{21}} = \frac{5}{5} = 1$$

b) $y_{22}^{\bar{B}} = \frac{a_{22}}{a_{21}} = \frac{0}{5} = 0$

$$y_{23}^{\bar{B}} = \frac{a_{23}}{a_{21}} = -\frac{1}{2} \div 5 = -\frac{1}{10}$$

$$y_{24}^{\bar{B}} = \frac{a_{24}}{a_{21}} = \frac{1}{5}$$

- c) Pentru determinarea primului rând $i=1$ din tabelul T_2 se calculează:

$$\bar{x}_i^B = \bar{x}_i - y_{ik} \cdot \frac{\bar{x}_1}{y_{1k}}$$

$$\bar{x}_2^B = \bar{x}_2 - a_{11} \cdot \frac{\bar{x}_4}{a_{21}} = 10 - \frac{1}{2} \cdot \frac{40}{5} = 6;$$

$$\bar{y}_{ij}^B = \bar{y}_{ij} - y_{ik} \cdot \frac{y_{1j}}{y_{1k}}$$

$$\bar{a}_{12}^B = \bar{a}_{12} - \frac{a_{22} \cdot a_{11}}{a_{21}} = 1 - 0 \cdot \frac{1}{2} \cdot \frac{1}{5} = 1$$

$$\bar{a}_{13}^B = \bar{a}_{13} - \frac{a_{23} \cdot a_{11}}{a_{21}} = \frac{1}{4} + \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{5} = \frac{1}{4} + \frac{1}{20} = \frac{6}{20} = \frac{3}{10}$$

$$\bar{a}_{14}^B = \bar{a}_{14} - \frac{a_{24} \cdot a_{11}}{a_{21}} = 0 - 1 \cdot \frac{1}{2} \cdot \frac{1}{5} = -\frac{1}{10}$$

- d) Indicatorii \bar{z}_B și $\bar{z}_j^B - c_j$ se calculează cu relațiile:

$$\bar{Z}^B = \sum_{i \in J^B} c_i \bar{x}_i = 15 \cdot 6 + 10 \cdot 8 = 90 + 80 = 170$$

$$\bar{z}_1^B = \sum_{i \in J^B} c_i a_{i1} = c_1 a_{11} + c_2 a_{21} = 15 \cdot 0 + 10 \cdot 1 = 10$$

$$\bar{z}_2^B = \sum_{i \in J^B} c_i a_{i2} = c_1 a_{12} + c_2 a_{22} = 15 \cdot 1 + 10 \cdot 0 = 15$$

$$\bar{z}_3^B = \sum_{i \in J^B} c_i a_{i3} = c_1 a_{13} + c_2 a_{23} = 15 \cdot \frac{3}{10} + 10 \cdot \frac{1}{10} = \frac{45}{10} - 1 = \frac{35}{10} = \frac{7}{2}$$

$$\bar{z}_4^B = \sum_{i \in J^B} c_i a_{i4} = c_1 a_{14} + c_2 a_{24} = 15 \cdot \frac{1}{10} + 10 \cdot \frac{1}{5} = \frac{1}{2}$$

$$\Delta z_1 = z_1^{\bar{B}} - c_1 = 10 - 10 = 0$$

$$\Delta z_2 = z_2^{\bar{B}} - c_2 = 15 - 15 = 0$$

$$\Delta z_3 = z_3^{\bar{B}} - c_3 = \frac{7}{2} - 0 = \frac{7}{2}$$

$$\Delta z_4 = z_4^{\bar{B}} - c_4 = \frac{1}{2} - 0 = \frac{1}{2}$$

Exemplul 2. Să considerăm problema de programare liniară:

$$\begin{cases} \min[z(x)] = x_1 + 6x_2 \\ 2x_1 + x_2 \geq 3 \\ x_1 + 3x_2 \geq 4 \\ x_j \geq 0 (j=1,2) \end{cases}$$

Introducând variabilele ecart x_3 și x_4 obținem forma standard:

$$\begin{cases} \min[z(x)] = x_1 + 6x_2 \\ 2x_1 + x_2 - x_3 = 3 \\ x_1 + 3x_2 - x_4 = 4 \\ x_j \geq 0 (j=1,2,3,4) \end{cases}$$

Matricea $A = (a^1, a^2, a^3, a^4) = \begin{pmatrix} 2 & 1 & -1 & 0 \\ 1 & 3 & 0 & -1 \end{pmatrix}$ are rangul 2.

Vectorii coloană corespunzători variabilelor ecart (adică vectorii a^3 și a^4) formează o bază, care nu mai este însă admisibilă, deoarece soluția de bază corespunzătoare este $x_3 = -3, x_4 = -4, x_1 = x_2 = 0$ și nu verifică condițiile de nenegativitate.

Se poate vedea însă că baza $B = (a^1, a^2)$ este admisibilă, adică poate fi luată drept bază de plecare. Pentru a alcătui tabelul simplex corespunzător acestei baze trebuie să calculăm $\bar{x} = B^{-1}b$ și $y_j^B = B^{-1}a^j$; aceste cantități apar la înmulțirea relației $Ax=b$ la stânga cu inversa bazei B , care este:

$$B^{-1} = \begin{pmatrix} 3/5 & -1/5 \\ -1/5 & 2/5 \end{pmatrix}$$

Scriind dezvoltat acest lucru obținem:

$$\begin{pmatrix} 3/5 & -1/5 \\ -1/5 & 2/5 \end{pmatrix} \begin{pmatrix} 2 & 1 & -1 & 0 \\ 1 & 3 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3/5 & -1/5 \\ -1/5 & 2/5 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

După efectuarea calculelor rezultă

$$x_1 - \frac{3}{5}x_3 + \frac{1}{5}x_4 = 1$$

$$x_2 + \frac{1}{5}x_3 - \frac{2}{5}x_4 = 1$$

Tabelul simplex corespunzător bazei $B = (a^1, a^2)$

| | | x_1 | x_2 | x_3 | x_4 |
|-------|---|-------|-------|-------|-------|
| | 7 | 0 | 0 | 3/5 | -11/5 |
| x_1 | 1 | 1 | 0 | -3/5 | 1/5 |
| x_2 | 1 | 0 | 1 | 1/5 | -2/5 |

Iterația 1. Deoarece există indici j pentru care avem $z_j - c_j > 0$ (problema este de minimizare) rezultă că programul de bază corespunzător nu este optim. Criteriul de intrare arată că vectorul care va intra în bază este α^3 . Aplicând criteriul de ieșire din bază rezultă că vectorul α^2 părăsește baza. Efectuând transformarea tabelului simplex precedent (pivotal este $y_{23} = 1/5$) obținem tabelul simplex pentru noua bază:

Tabelul simplex corespunzător bazei $B = (\alpha^1, \alpha^3)$

| | | x_1 | x_2 | x_3 | x_4 |
|-------|---|-------|-------|-------|-------|
| | 4 | 0 | -3 | 0 | -1 |
| x_1 | 4 | 1 | 3 | 0 | -1 |
| x_2 | 5 | 0 | 5 | 1 | -2 |

Deoarece toți $z_j - c_j$ sunt nepozitivi rezultă că am obținut programul optim $x_1 = 4, x_2 = 0, x_3 = 5, x_4 = 0$.

În figura 12 este reprezentată mulțimea soluțiilor primei probleme de programare liniară considerată aici; tronsonul programelor este în acest caz mulțimea convexă nemărginită determinată de segmentele AB și BC și semidreptele Ax_2 și Cx_1 . Este clar că iterația făcută se poate interpreta geometric ca o deplasare din vârful B până în vârful C pe muchia BC.

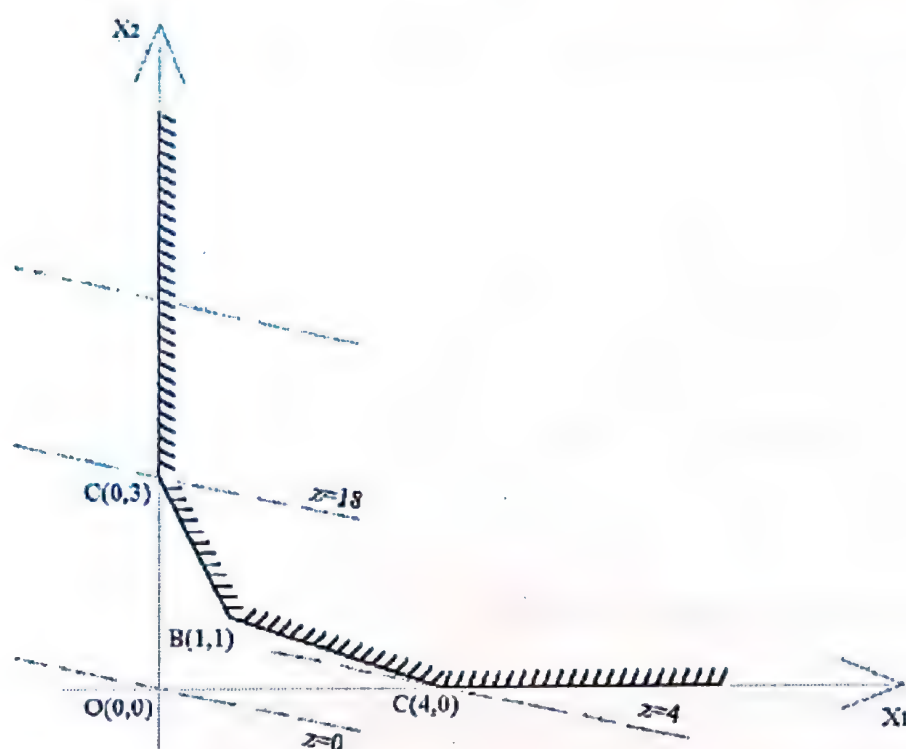


Figura 12

Exemplul 3. Să considerăm acum problema determinării maximului, rămânând valabile restricțiile problemei de la exemplul 2:

$$\begin{cases} \max[z(x)] = x_1 + 6x_2 \\ 2x_1 + x_2 \geq 3 \\ x_1 + 3x_2 \geq 4 \\ x_j \geq 0 (j = 1, 2) \end{cases}$$

Luînd drept bază inițială tot pe $B = (a^1, a^2)$ și utilizând aceeași observație ca mai sus obținem primul tabel simplex:

Tabelul simplex corespunzător bazei $B = (a^1, a^3)$

| | | x_1 | x_2 | x_3 | x_4 |
|-------|---|-------|-------|--------|---------|
| | 7 | 0 | 0 | $3/5$ | $-11/5$ |
| x_1 | 1 | 1 | 0 | $-3/5$ | $1/5$ |
| x_2 | 1 | 0 | 1 | $1/5$ | $-2/5$ |

Deoarece există indici $z_j - c_j < 0$ (problema este de maximizare) rezultă că programul de bază corespunzător nu este optim. Criteriul de intrare arată că vectorul care va intra în bază este a^4 . Aplicând criteriul de ieșire din bază rezultă că vectorul a^1 părăsește baza. Efectuând transformarea tabelului simplex precedent (pivotul este $y_{14} = 1/5$) obținem tabelul simplex pentru noua bază:

Tabelul simplex corespunzător bazei $B = (a^4, a^2)$

| | | x_1 | x_2 | x_3 | x_4 |
|-------|----|-------|-------|-------|-------|
| | 18 | 11 | 0 | -6 | 0 |
| x_1 | 5 | 5 | 0 | -3 | 1 |
| x_2 | 3 | 2 | 1 | -1 | 0 |

Deoarece există indici $z_j - c_j < 0$ rezultă că programul corespunzător bazei $B = (a^4, a^2)$ nu este optim. Criteriul de intrare arată că vectorul care va intra în bază este a^3 . Deoarece toți y_{i3} sunt negativi ($y_{43} = -3, y_{23} = -1$) rezultă că problema are optim infinit. Acest rezultat se putea obține în acest caz, observând direct pe figura 12 că dreptele $x_1 + 6x_2 = z$ întâlnesc tronsonul programelor pentru valori oricât de mari ale lui z .

CAPITOLUL IV**PROGRAME PENTRU REZOLVAREA PROBLEMELOR DE
PROGRAMARE LINIARĂ**

În baza algoritmului simplex primal descris în paragraful anterior, s-a întocmit un subprogram de tip subrutină denumit **SIMPLEX**. Acesta apelează în momente bine determinate alte trei subprograme **COLPIV**, **LINPIV** și **PIVOT** în scopul efectuării unor sarcini specifice, așa cum va fi explicat mai departe. În aceste subprograme nu sunt prevăzute nici un fel de instrucțiuni pentru operații de intrare-ieșire. Transmiterea datelor inițiale, precum și retransmiterea rezultatelor se face de la, respective către, un program principal prin argumente.

Argumentele care fac legătura la subprogramul **SIMPLEX** au semnificația descrisă în continuare.

N reprezintă numărul variabilelor independente de care depind funcția obiectiv și relațiile de restricții.

M reprezintă numărul variabilelor dependente, egal cu numărul total al restricțiilor, din care:

M1 inegalități de tipul $a_{11}x_1 + a_{12}x_2 + \dots + a_{1m}x_m \leq b_1$ scrise sub

forma $b_i + \sum_{k=1}^n a_{ik}x_k \geq 0; b_i \geq 0,$

M2 inegalități de tipul $a_{k1}x_1 + a_{k2}x_2 + \dots + a_{km}x_m \geq b_k$, scrise sub

forma $b_i + \sum_{k=1}^n a_{ik}x_k \leq 0; b_i \geq 0$;

M3 restricții de egalitate, scrise sub forma

$b_i + \sum_{k=1}^n a_{ik}x_k = 0; b_i \geq 0$.

A este tabloul simplex, corespunzător problemei programării liniare; acesta cuprinde coeficienții restricțiilor, precum și ai funcției obiectiv, după cum urmează:

M1 restricții de forma $b_i + \sum_{k=1}^n a_{ik}x_k \geq 0; b_i \geq 0$ pe liniile 2 la

M1+1;

M2 restricții de forma $b_i + \sum_{k=1}^n a_{ik}x_k \leq 0; b_i \geq 0$ pe liniile **M1+2**

la **M1+M2+1**;

M3 restricții de forma $b_i + \sum_{k=1}^n a_{ik}x_k = 0; b_i \geq 0$ pe liniile

M1+M2+2 la **M+1**.

Funcția obiectiv și restricțiile se scriu în așa fel ca termenii liberi să apară pe prima coloană. Se rezervă în tabloul **A** o linie în care se va forma, desigur numai în ipoteza necesității, funcția obiectiv artificială, conform celor descrise de funcția $F' = 2x_1 + 3x_2$. Pentru o mai ușoară manevrare a coeficienților, tabloul simplex este transformat într-un tablou cu o singură dimensiune, utilizând:

- **PL**, pasul de linie, mărime care reprezintă distanța în memorie dintre două elemente consecutive ale aceleiași coloane;
- **PC**, pasul de coloană, mărime care reprezintă distanța în memorie dintre două elemente consecutive ale aceleiași linii;

În aceste condiții, dacă facem citirea coeficienților pe linie vom avea $PL=N+1$ și $PC=1$, iar dacă facem citirea coeficienților pe coloană vom avea $PL=1$ și $PC=M+1$.

- **K** este dimensiunea tabloului **A**, având în urma celor menționate, valoarea $(M+2)*(N+1)$.
- **L1** este un tablou de dimensiune **N** care conține în fiecare moment indicii coloanelor admise la interschimburi; **L10** este numărul acestor coloane.
- **L2** este un tablou de dimensiune **M** care conține în fiecare moment indicii liniilor admise la interschimburi; **L20** este numărul acestor linii.
- **L3** este un tablou de dimensiune **M**, din care numai **M2** elemente sunt active, și specifică prin valoarea 1 a acestor elemente, câte restricții de forma

$$b_i + \sum_{k=1}^n a_{ik} x_k \leq 0; b_i \geq 0$$
 au mai rămas a fi tratate.
- **K1** este un tablou de dimensiune **N** care conține indicii variabilelor din afara bazei.

- **K2** este un tablou de dimensiune **M** care conține indicii variabilelor bazei.
- **EPS** este o mărime care exprimă precizia cu care vrem să obținem rezultatele. Valoarea sa trebuie aleasă cu foarte multă atenție.
- **IER** este o mărime care după executarea programului furnizează informații asupra corectitudinii rezultatelor și anume:
 - pentru **IER=0** soluția problemei este corectă, cazul tratat are un optim;
 - pentru **IER=1** problema nu are o soluție finită;
 - pentru **IER=2** nu există o soluție admisibilă de bază;

La începutul subprogramului se atribuie variabilei de tip întreg **R** valoarea 0. Rolul acesteia constă în a indica pentru cazul analizat, etapa în care ne aflăm și anume: dacă **R** este 0 s-a obținut prima soluție admisibilă de bază, dacă **R** este 1 nu s-a obținut încă această soluție.

Prin următoarele instrucțiuni tablourile **L1**, **L2**, **K1** și **K2** primesc valorile corespunzătoare punctului de start. De reținut că, pentru starea inițială, variabilele bazei au indicii **N+1** la **M**, iar variabilele din afara bazei indicii 1 la **N**. Această stare formează prima soluție admisibilă de bază, numai dacă restricțiile sunt de forma

$$b_i + \sum_{k=1}^n a_{ik} x_k \geq 0; b_i \geq 0, \text{ adică pentru cazul în care } M2=0 \text{ și } M3=0. \text{ Prin}$$

urmare, în ipoteza îndeplinirii condiției $M2+M3=0$, și cum R are valoarea 0 dată inițial, se face saltul la instrucțiunea cu eticheta 190 pentru a căuta o soluție admisibilă de bază mai bună, în sensul optimizării funcției obiectiv.

Se apelează subprogramul **COLPIV** care determină coloana pivot, returnând programului apelant pe T , indicele coloanei pivot și **AMAX** valoarea coeficientului corespunzător coloanei pivot din funcția obiectiv. De menționat că, datorită modului în care au fost scriși coeficienții în tabloul A , precum și exemplele analizate, **AMAX** este coeficientul cu cea mai mare valoare pozitivă din funcția obiectiv.

Pentru a înlătura influența erorilor de rotunjire în corecta ramificare a programului, **AMAX** este adus la valoarea 0 dacă modulul său este inferior valorii **EPS**. În continuare, se testează **AMAX** față de 0. În momentul în care acesta este negativ, adică toți coeficienții funcției obiectiv sunt negativi (se va remarca din nou semnul inversat față de demonstrațiile teoretice, datorită modului de completare inițială a tabloului A), s-a ajuns la tabloul simplex optimal, parametrul **IER** este făcut egal cu 0 și rezultatele sunt transmise programului apelant. În caz contrar, se apelează subprogramul **LINPIV**, care determină linia pivot S . În continuare, dacă problema are o soluție finită se apelează subprogramul **PIVOT** care efectuează operația de pivotare. Se revine apoi la instrucțiunea cu eticheta 180, de la care, prin următoarele trei instrucțiuni, se fac modificările valorilor indicilor în tablourile $K1$ și $K2$

Adina Butnărașu

în sensul trecerii în bază a variabilei cu indicele T, precum și în afara bazei a variabilei cu indicele S.

În acest moment o iterație completă a fost efectuată. Se apelează din nou subprogramul **COLPIV** și în ipoteza în care **AMAX** este pozitiv se trece la o nouă iterație. Procedul descris continuă, efectuându-se atâtea iterații câte sunt necesare obținerii tabloului simplex optimal.

Dacă $M2+M3 \neq 0$, problema programării liniare necesită găsirea inițială a unei soluții de bază admisibilă. În astfel de situații, după instrucțiunea în care se compară $M2+M3$ cu 0, programul continuă cu ramura în care **R** ia valoarea 1, pentru a indica că nu s-a obținut prima soluție admisibilă de bază. Înaintea acestei instrucțiuni există două cicluri **DO** prin care se atribuie valori corespunzătoare elementelor tabloului **L3**. Astfel, ciclul **DO** cu eticheta finală 30 atribuie inițial pentru **M** elemente ale acestui tablou valoarea 0. Acest ciclu s-a introdus pentru a putea utiliza subprogramul **SIMPLEX** de către programul principal prin apeluri repetate. Ciclul **DO** menționat elimină eventualele valori neegale cu 0, păstrate în tabloul **L3** de la exemplele anterioare. Cel de al doilea ciclu **DO** atribuie, în ipoteza în care $M2 \neq 0$, valoarea 1 pentru $M2$ elemente ale tabloului **L3**.

Prin ciclul **DO** cu eticheta finală 70 se formează, în ultima linie a tabloului simplex, elementele de start ale funcției obiectiv artificiale. În momentul de start al procedurii simplex, funcția obiectiv artificială nu trebuie să conțină variabilele bazei (acest lucru este valabil și pentru

Programarea liniară. Algoritmul Simplex

punctul de start al problemei programării liniare pusă sub forma canonică admisibilă. În consecință, prin ciclul **DO** menționat, din elementele de start ale ultimei linii din tabloul simplex, care corespund funcției obiectiv artificiale, au fost eliminate variabilele bazei.

Prin instrucțiunea cu eticheta **80** se face apelul subprogramului **COLPIV** pentru a determina coloana pivot în etapa auxiliară de calcul a primei soluții admisibile de bază. În continuare, după o tehnică asemănătoare celei menționate anterior sunt apelate subprogramele **LINPIV** și **PIVOT**, în scopul efectuării unei iterații complete. Se verifică apoi indicele variabilei care urmează a părăsi baza. Dacă acesta corespunde unei restricții de forma

$$b_i + \sum_{k=1}^n a_{ik} x_k \geq 0; b_i \geq 0$$

se trece la instrucțiunea cu eticheta **180**, dacă corespunde unei restricții de forma

$$b_i + \sum_{k=1}^n a_{ik} x_k \leq 0; b_i \geq 0$$

elementul din tabloul **L3** este făcut egal cu **0**, iar dacă corespunde unei restricții de forma

$$b_i + \sum_{k=1}^n a_{ik} x_k = 0; b_i \geq 0$$

se asigură ca această variabilă să rămână în afara bazei. În cazul în care **K2(S)** corespunde unei restricții de forma

$$b_i + \sum_{k=1}^n a_{ik} x_k \leq 0; b_i \geq 0$$

sau

$$b_i + \sum_{k=1}^n a_{ik} x_k = 0; b_i \geq 0,$$

prin ciclul **DO** cu eticheta finală **170** se calculează coeficienții **A(J)** aferenți. Urmează, cu începere de la instrucțiunea cu eticheta **180**, o schimbare a valorilor indicilor din tablourile **K1** și **K2** și o revenire la instrucțiunea cu eticheta **80** pentru o nouă iterație, atât timp cât parametrul **R** are valoarea **1**. În momentul în care s-a ajuns la optimul funcției obiectiv auxiliare, și acest optim este **0** (pus în evidență în instrucțiunea cu eticheta **90** prin **A(J)**), parametrul **R** ia valoarea **0** și se efectuează saltul la instrucțiunea cu etichete **190** pentru a efectua cea de a doua etapă, respectiv calculul optimului funcției obiectiv, dată prin problema în studiu.

Subprogramul **COLPIV** determină coloana pivot pentru fiecare iterație a algoritmului simplex. Parametrii care fac legătura de la programul apelant sunt: tabloul **A** de dimensiune **K**, tabloul **L1** de dimensiune **JL1**, mărimile de tip întreg **L10**, **IN**, **PL** și **PC**. Acești parametri au semnificația cunoscută, mai puțin mărimea **IN**, care specifică linia în care este plasată funcția obiectiv, și anume:

- dacă funcția obiectiv se află pe prima linie, cazul funcției obiectiv al cărui optim trebuie căutat, **IN** este egal cu **0**;

- dacă funcția obiectiv se află pe ultima linie, cazul funcției obiectiv artificiale, Ineste egal cu $M+1$.

Parametrii calculați de către subprogram sunt **AMAX**, cel mai mare coeficient pozitiv al funcției obiectiv și **T**, indicele coloanei pe care se află acest coeficient, adică coloana pivot. Determinarea coeficientului **AMAX** și a mărimii **T** se face clasic, folosind algoritmul determinării celui mai mare număr dintr-o mulțime dată. Pentru acest scop se folosește un ciclu **DO** prin care în prealabil au fost stabiliți cu ajutorul mărimilor **IN**, **PL** și **PC** indicii elementelor tabloului **A** care urmează a fi comparați.

Subprogramul **LINPIV** determină linia pivot pentru fiecare iterație a algoritmului simplex. Parametrii care fac legătura de la programul apelant sunt: tabloul **A** de dimensiune **K**, tabloul **L2** de dimensiune **JL2**, mărimile de tip întreg **L20**, **PL**, **PC**, **N** și **T**. Toți acești parametri au semnificația cunoscută. În plus apare parametrul **V** care poate avea valoarea 1 sau -1, în cazul de față -1. Subprogramul calculează indicele liniei pivot **S**. Inițial se atribuie mărimii de tip întreg **S** valoarea 0. Prin primul ciclu **DO** cu eticheta finală 10 se determină primul coeficient al coloanei pivot care este negativ. Apoi, prin ciclul **DO** cu eticheta finală 60 sunt analizați acești coeficienți pentru a alege

indicele liniei care conduce la $\min \left\{ \frac{b_i}{a_{iu}} \right\}, i = \overline{1, m}$. Dacă pentru doi

coeficienți se ajunge la aceeași valoare a raportului $\frac{b_i}{a_{iu}}$ ne aflăm în

cazul de degenerare. În subprogramul propus se caută eliminarea cazului de degenerare alegându-se pentru astfel de situații indicele liniei

care conduce la $\min \left\{ \frac{a_{ih}}{a_{hh}} \right\}, \quad i = \overline{1, m}$. Această tehnică permite

continuarea algoritmului simplex, și în majoritatea cazurilor practice la eliminarea deficienței impuse de degenerare.

Subprogramul **PIVOT** efectuează operația de pivotare pentru fiecare iterație a algoritmului simplex. Parametrii care fac legătura de la programul apelant sunt: tabloul **A** de dimensiune **K**; **I0** linia cu care începe și **I1** linia până la care se efectuează operația de pivotare; **JO** coloana cu care începe și **J1** coloana până la care se efectuează operația de pivotare; **PL**, **PC**, **S** și **T** mărimide tip întreg cu semnificația cunoscută. Subprogramul înlocuiește valorile coeficienților tabloului **A** în urma operației de pivotare.

Inițial, se determină poziția elementului pivot. În continuare, prin ciclurile **DO** suprapuse cu eticheta finală **20** se efectuează operația efectivă de pivotare, care constă în înlocuirea fiecărui termen al tabloului **A**, mai puțin termenii de pe linia pivot, respectiv coloana pivot. Subprogramul se încheie prin ciclul **DO** cu eticheta finală **30**, prin care se calculează noii termeni ai liniei pivot.

**Exemple specifice ale unor probleme de programare liniară
rezolvate cu ajutorul subprogramului simplex**

Important este faptul că apelul subprogramului **SIMPLEX** propus, se face de către un program principal. Un astfel de program prezentăm mai jos.

Curățirea sa nu pune nici un fel de probleme. Alături de parametrii cunoscuți care trebuie citați, apare și mărimea de tip întreg **IM**. Aceasta specifică prin valoarea sa natura problemei de optimizare și anume:

- pentru **IM=0** se caută maximul funcției obiectiv în prezența relațiilor de restricții date;
- pentru **IM \neq 0** se caută minimul funcției obiectiv în prezența relațiilor de restricții date.

În consecință, sunt introduse instrucțiuni prin care, în ipoteza **IM \neq 0**, se modifică inițial semnul coeficienților funcției obiectiv, precum și în final semnul valorii optimului obținut.

S-a prevăzut imprimarea tabloului **SIMPLEX** inițial, precum și după returnarea rezultatelor, a codului pentru semnalarea erorii, a valorii optime pentru funcția obiectiv și a valorilor variabilelor pentru care s-a obținut acest optim. Autorul recomandă imprimarea codului pentru semnalarea erorii, deoarece numai în ipoteza în care valoarea sa este 0, rezultatele pot fi considerate corecte. De fapt, pentru valori diferite de 0

Adina Butnărașu

ale acestui parametru nici nu are sens a imprima rezultatele returnate de subprogramul **SIMPLEX**. Dacă utilizatorul dorește a imprima în mod succesiv după fiecare iterație tabloul simplex, precum și indicii variabilelor bazei și ai celor din afara bazei, este necesar a introduce instrucțiuni de ieșire în locurile corespunzătoare ale subprogramului **SIMPLEX**, de exemplu înaintea instrucțiunii **IF** care compară parametrul **R** cu 0.

Programul principal, în forma prezentată, poate rezolva în mod repetat, exemple de probleme de programare liniară în care funcția obiectiv și relațiile de restricții sunt de trei variabile. Efectuând modificări minore în instrucțiunile de intrare-ieșire, în special în instrucțiunile **FORMAT**, programul principal propus poate fi utilizat pentru situații în care numărul variabilelor diferă de trei.

Exemplu 1: Se cere maximizarea funcției obiectiv

$$F=3x_1 + 5x_2 + 4x_3,$$

În prezența restricțiilor

$$\begin{cases} 2x_1 + 3x_2 \leq 8 \\ 2x_2 + 5x_3 \leq 10 \\ 3x_1 + 3x_2 + 4x_3 \leq 16 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Parametrii care trebuie citați în programul principal sunt: **N=3**, **M1=3**, **M2=0**, **M3=0**, **PL=4** (citirea se face pe linie); **PC=1**, **EPS=10⁻⁴**,

IM=0, K=20, precum și coeficienții tabloului simplex. Rezultatele obținute la imprimantă sunt prezentate mai jos.

Exemplul 2: Se cere minimizarea funcției obiectiv:

$$F=6x_1 + 2x_2 + 3x_3,$$

în prezența restricțiilor

$$\begin{cases} 30x_1 + 20x_2 + 40x_3 \geq 34 \\ x_1 + x_2 + x_3 = 1 \\ 10x_1 + 70x_2 \leq 11 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Parametrii care trebuie citați în programul principal sunt: **N=3, M1=1, M2=1, M3=1, PL=4, PC=1, EPS=10⁻³, IM=1, K=20**, precum și coeficienții tabloului simplex. Rezultatele obținute la imprimantă sunt prezentate mai jos.

Dacă exemplul propus îl rulăm cu aceleași date, modificând numai valoarea mărimii **EPS=10⁻⁶**, obținem la imprimantă rezultatele prezentate în următorul tabel:

Cu valoarea **2** pentru codul de semnalare a erorii. Din analiza acestui exemplu rezultă importanța alegerii celei mai adecvate valori pentru parametrul **EPS**.

O valoare recomandată [13] pentru acest parametru trebuie să respecte inegalitatea

$$\varepsilon \geq \frac{\sum |a_{ij}|}{k} \cdot 10^{(-m+1)}, \text{ în care:}$$

- $\sum |a_{ij}|$ reprezintă suma valorilor absolute ale coeficienților matricei A;
- k numărul coeficienților matricei A;
- m numărul de cifre semnificative cu care calculatorul lucrează în virgulă mobilă.

Mai jos se prezintă imprimările **programului sursă Simplex**:


```

5000 REM Program Simplex Dual
5005 DIM G(2): G(1) = 1: X1 = 1: INDCT7 = 1: INDCTR = 1: ICHTR = 0:
IOUT1 = 0: I1ROW = 2000: DELTAA = 9.999999E-21: IOUT2 = 1: IOUT3 = 1
5010 COLOR 0, 3: CLS: PRINT "PROGRAM SimplexD pentru programare
lineara ***Algoritmul Simplex Dual ***"
5015 VS =
-----
": PRINT VS: ILPRINT = 0: INPUT "DORITI LISTAREA
DATELOR LA IMPRIMANTA      da/nu      "; W$: IF W$ = "da" THEN ILPRINT
= 1
5020 IF ILPRINT = 1 THEN LPRINT "PROGRAM SimplexD pentru Programare
lineara      ***Algoritmul Simplex Dual ***"
5025 INPUT "Problema de: 1 Minim      -1 Maxim      "; TEXTR
5030 PRINT SPC(16); "Cod extrem = "; TEXTR: IF ILPRINT = 1 THEN
LPRINT "Cod extrem (1 pentru minim, -1 pentru maxim) = "; TEXTR
5035 INPUT "Numar variabile = "; NV
5040 IF ILPRINT = 1 THEN LPRINT "Numar variabile="; NV
5045 N = NV + 1: N1 = N + 1: NM1 = N - 1: NO = N - 1
5050 INPUT "Numar restrictii = "; MR
5055 IF ILPRINT = 1 THEN LPRINT "Numar restrictii="; MR
5060 M = MR + 1: M1 = M + 1: MO = M - 1
5065 DIM A(M1, N): DIM U(N1): DIM T(N): DIM R(M): DIM C(N)
5070 PCTOL = 0: SOLMIN = 0: R(1) = 0
5075 FOR I = 1 TO N: U(I) = 0: NEXT I
5080 INPUT "Doriti citirea coef. sistemului dintr-un fisier      da/nu
"; W$
5085 IF W$ = "nu" OR W$ = "NU" THEN GOTO 5125
5090 INPUT "Numele complet al fisierului "; NF$
5095 OPEN "I", #1, NF$
5100 FOR I = 1 TO M
5105 FOR j = 1 TO N
5110 INPUT #1, A(I, j)
5115 NEXT j
5120 NEXT I: CLOSE #1: GOTO 5270
5125 PRINT "Coeficientii functiei obiectiv:"; A(1, 1) = 0
5130 I = 1: FOR j = 2 TO N: PRINT "Coloana "; j - 1; : INPUT "
Val.coef.="; A(I, j): A(I, j) = A(I, j) * TEXTR: NEXT j
5135 IF ILPRINT = 1 THEN LPRINT "Coeficientii functiei obiectiv:";
FOR j = 2 TO NV + 1: LPRINT "COLOANA "; j - 1; " VAL.COEF.="; A(1,
j): NEXT j
5140 PRINT : PRINT "Coeficientii restrictiilor, tipurile de
restrictii si termenii liberi:"
5145 PRINT : Z$ = "-----": FOR I = 2 TO M: PRINT
"Restrictia nr. "; I - 1: PRINT Z$
5150 IF ILPRINT = 1 THEN LPRINT "Restrictia nr.      "; I - 1: LPRINT Z
$
5155 PRINT "Coeficientii restrictiei, <>0 (0 pentru terminare
intrare)": PRINT
5160 IF ILPRINT = 1 THEN LPRINT "Coeficientii restrictiei, <>0:"
5165 INPUT "Coloana nr. "; j: IF j = 0 THEN GOTO 5180
5170 INPUT " Val.coef.="; A(I, j + 1): IF ILPRINT = 1 THEN LPRINT
"Coloana nr. "; j: " Valoare coef.="; A(I, j + 1)
5175 PRINT : GOTO 5165
5180 INPUT "Tip restrictie <= = => "; W$
5185 IF W$ = "<=" OR W$ = "<" THEN LET R(I) = -1: GOTO 5205
5190 IF W$ = "=" THEN LET R(I) = 0: GOTO 5205
5195 IF W$ = ">=" OR W$ = ">" THEN LET R(I) = 1: GOTO 5205
5200 PRINT "Eroare tip restrictie !!! Reintroduceti corect tipul de
restrictie !": GOTO 5180
5205 PRINT "Tip "; W$; " Cod "; R(I)
5210 IF ILPRINT = 1 THEN LPRINT "Tip restrictie"; W$; " Cod "; R(I)
5215 INPUT "Termenul liber = "; A(I, 1): PRINT VS: IF ILPRINT = 1
THEN LPRINT "Termenul liber = "; A(I, 1): LPRINT VS
5220 NEXT I
5225 INPUT "Doriti salvarea coef. sistemului intr-un fisier      da/nu
"; W$
5230 IF W$ = "nu" OR W$ = "NU" THEN GOTO 5270
5235 INPUT "Numele complet al fisierului "; NF$

```

```

5240 OPEN "O", #1, NFS
5245 FOR I = 1 TO M
5250 FOR j = 1 TO N
5255 PRINT #1, A(I, j)
5260 NEXT j
5265 NEXT I: CLOSE #1
5270 IF M < 2 THEN GOTO 5295
5275 FOR I = 2 TO M: IF R(I) < 0 THEN GOTO 5285
5280 FOR j = 2 TO N: LET A(I, j) = -A(I, j): NEXT j: GOTO 5290
5285 A(I, 1) = -A(I, 1)
5290 NEXT I
5295 FOR I = 2 TO N: IF U(I) <= 0 THEN LET U(I - 1) = 2000
5300 NEXT I: FOR I = 2 TO N: LET C(I - 1) = 0: NEXT I
5305 IF M < 2 THEN GOTO 5325
5310 FOR I = 2 TO M: IF R(I) = 0 THEN GOTO 5320
5315 LET R(I) = 1 - I
5320 NEXT I
5325 A(1, 1) = A(1, 1): C(1) = 0
5330 FOR j = 2 TO N: IF A(1, j) >= 0 THEN GOTO 5340
5335 FOR I = 1 TO M: A(I, 1) = A(I, 1) + A(I, j) * U(j - 1): A(I, j)
= -A(I, j): NEXT I: C(j) = 2000 + j - 1: GOTO 5345
5340 C(j) = j - 1
5345 NEXT j
5350 GOTO 5565
5355 AMAX = 0: IF M < 2 THEN GOTO 5380
5360 FOR I = 2 TO M: IF A(I, 1) <= 0 THEN GOTO 5375
5365 IF A(I, 1) - AMAX <= 0 THEN GOTO 5375
5370 AMAX = A(I, 1): IPVR = I
5375 NEXT I
5380 IF AMAX <= 0 THEN GOTO 5595
5385 AMAX = -1E+35
5390 IF N - 2 >= 0 THEN GOTO 5400
5395 GOTO 5445
5400 IPVC = 0
5405 FOR j = 2 TO N
5410 IF A(IPVR, j) >= 0 THEN GOTO 5440
5415 RTIO = A(1, j) / A(IPVR, j): IF RTIO - AMAX < 0 THEN GOTO 5440
5420 IF RTIO - AMAX = 0 THEN GOTO 5435
5425 AMAX = RTIO
5430 IPVC = j: GOTO 5440
5435 IF A(IPVR, j) - A(IPVR, IPVC) < 0 THEN GOTO 5430
5440 NEXT j
5445 IF IPVC <> 0 THEN GOTO 5460
5450 PRINT "Problema infeasibila": IF ILPRINT = 1 THEN LPRINT
"Problema infeasibila"
5455 STOP
5460 ALFA = A(IPVR, IPVC)
5465 FOR j = 1 TO N: IF A(IPVR, j) = 0 THEN GOTO 5510
5470 IF j - IPVC = 0 THEN GOTO 5510
5475 ARTIO = A(IPVR, j) / ALFA
5480 FOR I = 1 TO M: IF A(I, IPVC) = 0 THEN GOTO 5505
5485 IF I - IPVR = 0 THEN GOTO 5505
5490 A(I, j) = A(I, j) - ARTIO * A(I, IPVC)
5495 IF ABS(A(I, j) - DELTAA) >= 0 THEN GOTO 5505
5500 A(I, j) = 0
5505 NEXT I
5510 NEXT j
5515 FOR j = 1 TO N: A(IPVR, j) = A(IPVR, j) / ALFA: NEXT j
5520 ISV = R(IPVR): R(IPVR) = C(IPVC)
5525 IF ISV <> 0 THEN GOTO 5540
5530 FOR I = 1 TO M: A(I, IPVC) = A(I, N): NEXT I
5535 C(IPVC) = C(N): N = N - 1: GOTO 5550
5540 FOR I = 1 TO M: A(I, IPVC) = -A(I, IPVC) / ALFA: NEXT I
5545 C(IPVC) = ISV: A(IPVR, IPVC) = 1 / ALFA
5550 PRINT "Iteratia "; ICNTR: ICNTR = ICNTR + 1: IF R(IPVR) + 2000
= 0 THEN GOTO 5560
5555 GOTO 5565
5560 FOR j = 1 TO N: A(IPVR, j) = A(M, j): R(IPVR) = R(M): M = M -

```



```

1: NEXT j
5565 IF M < 2 THEN GOTO 5590
5570 FOR K = 2 TO M: IF R(K) <> 0 THEN GOTO 5585
5575 IF A(K, 1) >= 0 THEN GOTO 5585
5580 FOR L = 1 TO N: A(K, L) = -A(K, L): NEXT L
5585 NEXT K
5590 GOTO 5355
5595 IF M < 2 THEN GOTO 5660
5600 FOR I = 2 TO M: IF R(I) <= 0 THEN GOTO 5655
5605 j = R(I): IF j - 2000 <= 0 THEN GOTO 5615
5610 j = j - 2000
5615 IF U(j) + A(I, j) >= 0 THEN GOTO 5655
5620 IF DELTAA + U(j) + A(I, 1) >= 0 THEN GOTO 5650
5625 A(I, 1) = -A(I, 1) - U(j)
5630 FOR K = 2 TO N: A(I, K) = -A(I, K): NEXT K
5635 IPVR = 1: IF j - R(I) = 0 THEN GOTO 5645
5640 R(I) = j: GOTO 5385
5645 R(I) = R(I) + 2000: GOTO 5385
5650 A(I, 1) = -U(j)
5655 NEXT I
5660 IF M < 2 THEN GOTO 5690
5665 FOR I = 2 TO M: IF R(I) <= 0 THEN GOTO 5685
5670 IF R(I) - 2000 <= 0 THEN GOTO 5680
5675 j = R(I) - 2000: T(j) = U(j) + A(I, 1): GOTO 5685
5680 j = R(I): T(j) = -A(I, 1)
5685 NEXT I
5690 FOR I = 2 TO N: IF C(I) <= 0 THEN GOTO 5710
5695 IF C(I) - 2000 <= 0 THEN GOTO 5705
5700 j = C(I) - 2000: T(j) = U(j): GOTO 5710
5705 j = C(I): T(j) = 0
5710 NEXT I
5715 ISF = 0: FOR I = 1 TO NM1: IF T(I) >= 2000 THEN LET ISF = 1
5720 NEXT I: IF ISF = 1 THEN PRINT "Solutie infinita"
5725 IF ISF = 1 AND ILPRINT = 1 THEN LPRINT "Solutie infinita": STOP
5730 IF ISF = 1 AND ILPRINT = 0 THEN STOP
5735 INPUT "Doriti afisarea solutiei optime da/nu "; WS
5740 IF WS <> "da" AND WS <> "DA" THEN STOP
5745 ZOPT = A(1, 1) * TEXTR: PRINT "Valoarea functiei obiectiv";
ZOPT
5750 IF ILPRINT = 1 THEN LPRINT "Valoarea functiei obiectiv="; ZOPT
5755 PRINT "Nr. iteratii efectuate"; ICNTR
5760 IF ILPRINT = 1 THEN LPRINT "Nr. iteratii efectuate"; ICNTR
5765 PRINT "Solutia optima"
5770 IF ILPRINT = 1 THEN LPRINT "Solutia optima"
5775 FOR I = 1 TO NM1
5780 PRINT "X("; I; ")="; T(I)
5785 IF ILPRINT = 1 THEN LPRINT "X("; I; ")="; T(I)
5790 NEXT I: INPUT stopf
5795 STOP

```


Problema 1:

$$\min(x_1 + 6x_2)$$

$$2x_1 + 3x_2 \geq 3$$

$$x_1 + 3x_2 \geq 4$$

$$x_1, x_2 \geq 0$$

Problema 1

Coefficientii restricției, <0> <0 pentru terminare intrare>

Coeficient nr. ? 1
Val.coef.=? 1

Coeficient nr. ? 2
Val.coef.=? 3

Coeficient nr. ? 0

Tip restricție: (< = > =) ? =)

Tip ? Cod 1

Termenul liber = ? 4

Doriti salvarea coef. sistemului intr-un fisier da/nu ? nu

Iteratia 0

Doriti afisarea solutiei optime da/nu ? da

Valoarea functiei obiectiv 1

Nr. iteratii efectuate 1

Solutia optima

X(1)= 4

X(2)= 0

?

Concluzie:

Soluțiile problemei sunt: $x_1 = 4$, $x_2 = 0$ iar valoarea funcției obiectiv este 4.

Problema 2:

$$\max(10x_1 + 15x_2)$$

$$2x_1 + 4x_2 + x_3 = 40$$

$$6x_1 + 2x_2 + x_4 = 60$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Problema 2

```

Coloana nr. 3
Val.coef.=7.0
Coloana nr. 4
Val.coef.=7.1
Coloana nr. 5
Tip restrictie <= - => 7 =
Tip = Cod 0
Termenul liber = 7.60
-----
Doriti salvarea coef. sistemului intr-un fisier da/nu ? nu
Iteratia 0
Iteratia 1
Doriti afisarea solutiei optime da/nu ? da
Valoarea functiei obiectiv 169.9981
Nr. iteratii efectuate 2
Solutie optima
X(1) = 7.999878
X(2) = 6
X(3) = 0
X(4) = 0
?
```

Concluzie:

Soluțiile problemei sunt: $x_1 = 7,999878$, $x_2 = 6$ iar valoarea funcției obiectiv este 169,9981.

Observație. Soluția exactă este $x_1 = 8$, $x_2 = 6$ iar valoarea funcției obiectiv este 170. Zecimalele apar datorită erorilor sistemului de calcul.

Problema 3:

$$\max(2x_1 + x_2)$$

$$x_1 - x_2 \leq 4$$

$$3x_1 - x_2 \leq 18$$

$$-x_1 + 2x_2 \leq 6$$

$$x_1, x_2 \geq 0$$

Problema 3

Coefficientii restricției. <0> <0 pentru terminare intrare>

Coloana nr. ? 1
Val.coef.=? -1

Coloana nr. ? 2
Val.coef.=? 2

Coloana nr. ? 0
Tip restricție <= = > ? <=
Tip <= Cod -1
Termenul liber = ? 6

Doriti salvarea coef. sistemului intr-un fisier da/nu ? nu

Iteratia 0

Iteratia 1
Doriti afisarea solutiei optime da/nu ? da

Valoarea functiei obiectiv 23.99948

Nr. iteratii efectuate 2

Solutia optima

X(1)= 8.399902

X(2)= 7.199829

?

Concluzie:

Soluțiile problemei sunt: $x_1 = 8,399902$, $x_2 = 7,199829$ iar valoarea funcției obiectiv este 23,99948.